

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 rozpoczynający się w roku akademickim

2023/2024

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: 5

Koordynator zajęć

Imię, nazwisko, tytuł/stopień naukowy, adres e-mail:

Monika Piróg, dr, monika.piróg@pwste.edu.pl

Jednostka organizacyjna: **Instytut 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:	15	Wykład:	
Ćwiczenia:		Ćwiczenia:	
Laboratorium:	40	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:	55	RAZEM:	

II. INFORMACJE SZCZEGÓŁOWE

Wymagania wstępne i dodatkowe:

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 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 i semantykę języków programowania, sposoby przekazywania parametrów do podprogramów, potrafi zdefiniować abstrakcyjne typy danych oraz wyjaśnić zasady przeciążania operatorów.		K_W03, K_W04
E_02	Zna i rozumie zastosowanie metodologii programowania obiektowego podczas rozwiązywania problemów informatycznych.		K_W03, K_W04
E_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.		K_W03, K_W04
Umiejętności - potrafi			
E_04	Potrafi pracować indywidualnie i w zespole; umie oszacować czas potrzebny na realizację zleconego zadania; potrafi opracować i zrealizować harmonogram prac zapewniający dotrzymanie terminów.		K_U01, K_U02
E_05	Ma umiejętność tworzenia projektów programistycznych w oparciu o języki programowania wysokiego poziomu.		K_U01, K_U02, K_U05, K_U18
E_06	Potrafi korzystać z dokumentacji i specyfikacji technicznych w celu dobrania odpowiednich parametrów i komponentów projektowanego systemu.		K_U01, K_U05
E_07	Potrafi sformułować algorytm, posługuje się językami programowania wysokiego poziomu oraz odpowiednimi narzędziami informatycznymi do opracowania programów komputerowych.		K_U01, K_U05, K_U18
Kompetencje społecznych - jest gotów do			
E_08	Rozumie potrzebę uczenia się przez całe życie.		K_K01, K_K02
E_09	Służy wsparciem i pomocą członkom zespołu w trudnych i stresujących sytuacjach.		K_K03, K_K04
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	<p>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.</p>		Wykład– prezentacja, wykład z dyskusją,	Test,
TP-02	<p>Wykorzystanie abstrakcyjnych typów danych. Możliwości wykorzystania przeciążania operatorów dla własnych typów danych. Przekazywanie argumentów do funkcji w poszczególnych językach programowania.</p>		Wykład– prezentacja, wykład z dyskusją,	Test,
TP-03	<p>Mechanizmy zarządzania pamięcią w języku C++, C# oraz Java. Przykłady programów. Omówienie cech charakterystycznych programowania imperatywnego.</p>		Wykład– prezentacja, wykład z dyskusją,	Test,

TP-04	<p>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#.</p>		Wykład– prezentacja, wykład z dyskusją,	Test,
TP-05	<p>Omówienie możliwości dziedziczenia i polimorfizmu w językach obiektowych. Pojęcie klasy bazowej i pochodnej, przesłanianie składowych, wiązanie statyczne i dynamiczne, funkcje wirtualne, klasy abstrakcyjne, interfejsy.</p>		Wykład– prezentacja, wykład z dyskusją,	Test,
TP-06	<p>Przykłady projektów realizujących zadania programistyczne wykorzystujących możliwości programowania obiektowego. Omówienie przykładów w wybranych językach C# i Java.</p>		Wykład– prezentacja, wykład z dyskusją,	Test

TP-07	<p>Programowanie graficznych interfejsów użytkownika. Przegląd komponentów oraz ich właściwości. Okna i rozmieszczenie komponentów. Zalety i wady ręcznego programowania interfejsu użytkownika.</p>		<p>Wykład– prezentacja, wykład z dyskusją,</p>	<p>Test,</p>
TP-08	<p>Możliwości szybkiego tworzenia graficznych interfejsów użytkownika w środowisku Eclipse, Microsoft Visual Studio.</p>		<p>Wykład– prezentacja, wykład z dyskusją,</p>	<p>Test,</p>
		laboratorium		
TP-09	<p>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.</p>		<p>wykład z dyskusją, realizacja zadań, praktyczna realizacja aplikacji</p>	<p>kolokwium, projekt zaliczeniowy</p>

TP-10	<p>Tworzenie i usuwanie tablic dynamicznych w języku C++.</p> <p>Pisanie programów oraz testowanie mechanizmu automatycznego zarządzania pamięcią w językach C# i Java.</p> <p>Tworzenie przykładowych programów wykorzystujących obiekty. Tworzenie schematów klas, konstruktorów, destruktorów.</p> <p>Pisanie funkcji składowych.</p> <p>Przekazywanie obiektów do funkcji w postaci argumentu. Funkcje zaprzyjaźnione oraz ich możliwości.</p> <p>Testowanie możliwości dostępu do składników obiektu, uruchamianie funkcji prywatnych.</p> <p>Przykłady w języku C i Java.</p>		wykład z dyskusją, realizacja zadań, praktyczna realizacja aplikacji	kolokwium, projekt zaliczeniowy
TP-11	<p>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.</p> <p>Dostęp do składników obiektów złożonych. Pisanie oraz testowanie funkcji wirtualnych.</p>		wykład z dyskusją, realizacja zadań, praktyczna realizacja aplikacji	kolokwium, projekt zaliczeniowy

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.		wykład z dyskusją, realizacja zadań, praktyczna realizacja aplikacji	kolokwium, projekt zaliczeniowy
TP-13	Tworzenie prostych aplikacji i apletów w języku Java. Układ graficzny aplikacji. Obsługa zdarzeń – interfejs nasłuchu zdarzeń. Pola wyboru, okna dialogowe.		wykład z dyskusją, realizacja zadań, praktyczna realizacja aplikacji	kolokwium, projekt zaliczeniowy
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.		wykład z dyskusją, realizacja zadań, praktyczna realizacja aplikacji	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.

ZALECANA LITERATURA (w tym pozycje w języku obcym)

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

1. Grębosz J.: Symfonia C++ standard, Edycja 2015
2. Snell M., Power L.: „Microsoft Visual Studio 2010, Helion, Gliwice, 2011,
3. Schildt H.: „Java : sztuka programowania” , Helion, Gliwice, 2004

Literatura uzupełniająca:

1. Boduch A.: Wstęp do programowania w C#, Helion, Gliwice, 2006
2. Troelsen S.: Język C# i Platforma .NET, Wydawnictwo Naukowe PWN, Warszawa 2006

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	60
Praca własna studenta	90
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: 5	2
	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 symbole 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

Ocena kształtująca: sposób pracy wykładowcy i studenta polega na systematycznym pozyskiwaniu informacji o przebiegu procesu uczenia się. Student otrzymuje informacje zwrotne dotyczące realizowanych zadań oraz projektu zaliczeniowego.

Ocena podsumowująca:

1. 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.
2. 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.
3. 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.

INFORMACJA O PRZEWIDYWANEJ MOŻLIWOŚCI WYKORZYSTANIA KSZTAŁCENIA NA ODLEGŁOŚĆ

.....
(data, podpis Koordynatora
odpowiedzialnego za zajęcia)

.....
(data, podpis Kierownika Zakładu/
Kierownika Jednostki Międzyinstytutowej)

Uwaga:
Karta opisu zajęć (syllabus) musi być dostępna dla studenta.