

Karta opisu zajęć - Sylabus

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

I. INFORMACJE PODSTAWOWE

Nazwa zajęć: Podstawy programowania	Cykl kształcenia rozpoczynający się w roku akademickim 2023/2024
Nazwa kierunku studiów, poziom i profil kształcenia: automatyka i elektronika praktyczna, pierwszego stopnia, profil praktyczny	
Język wykładowy: polski	Rodzaj zajęć: zajęcia kształcenia podstawowego
Rok studiów: I	Semestr: I
Liczba punktów ECTS przypisana zajęciom: 4	Koordinator zajęć Imię, nazwisko, tytuł/stopień naukowy, adres e-mail: Przemysław Hawro, dr inż., przemyslaw.hawro@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:		Laboratorium:	
Lektorat:		Lektorat:	
Projekt:		Projekt:	
Zajęcia praktyczne:	45	Zajęcia praktyczne:	
Seminarium:		Seminarium:	
Zajęcia terenowe:		Zajęcia terenowe:	
Praktyki zawodowe:		Praktyki zawodowe:	
Inna forma (jaka):		Inna forma (jaka):	
RAZEM:	60	RAZEM:	

II. INFORMACJE SZCZEGÓŁOWE

Wymagania wstępne i dodatkowe: podstawowe umiejętności matematyczne oraz informatyczne na poziomie szkoły ponadpodstawowej

Cel (cele) kształcenia dla zajęć: celem przedmiotu jest przedstawienie podstaw technik programowania opartych na właściwościach języka programowania C++, tj. podstawowe konstrukcje algorytmiczne, tworzenie i kompilacja programów, proste i złożone typy danych, programowanie proceduralne, operacje wejścia-wyjścia, pamięć operacyjna i wskaźniki.

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				
M_01	Zna podstawowe metody, techniki i narzędzia stosowane przy rozwiązywaniu prostych zadań informatycznych z zakresu teorii algorytmów, budowy systemów komputerowych, języków programowania.		K_W01, K_W04	
M_02	Charakteryzuje podstawowe pojęcia programistyczne: zadanie algorytmiczne, selekcja, iteracja, funkcja, rekurencja. Dobiera i stosuje podstawowe definicje algorytmiczno-programistyczne.		K_W04	
M_03	Dobiera metody projektowania oprogramowania zgodnie z metodyką strukturalną.		K_W04	
Umiejętności - potrafi				
M_04	Ustala kryteria integracji uzyskanych informacji z odpowiednią ich interpretacją.		K_U01, K_U18	
M_05	Wybiera metodę algorytmiczną do postawionego problemu oraz odpowiednie środowisko programistyczne.		K_U01, K_U18	
M_06	Opracowuje poszczególne etapy projektu oraz konstruuje optymalne rozwiązanie zadanego problemu.		K_U01, K_U02, K_U18	
Kompetencji społecznych - jest gotów do				
M_07	Jest otwarty na podnoszenie swoich kwalifikacji zawodowych poprzez uczestnictwo w szkoleniach specjalistycznych.		K_K01, K_K04	
M_08	Wykazuje odpowiedzialność za przestrzeganie zasad poszanowania nadrzędnej roli człowieka w szybko rozwijającej się technologii informacyjnej.		K_K02, 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.</p> <p>Pojęcie paradygmatu programowania strukturalnego.</p> <p>Struktura programu w języku C++, pliki źródłowe i nagłówkowe. Pojęcie algorytmu, przykłady podstawowych algorytmów w postaci schematów graficznych.</p>		wykład podający, analiza przykładów	Zaliczenie ustne, test
TP-02	<p>Wybrane środowiska programistyczne.</p> <p>Sposoby zapisu danych liczbowych w programie.</p> <p>Wykorzystanie zmiennych i stałych w programach.</p> <p>Rozwiązywanie prostych problemów algorytmicznych – rysowanie schematów blokowych oraz implementacja w kodzie języka C++.</p> <p>Wykorzystanie w programach wyrażeń arytmetycznych i logicznych.</p> <p>Zagadnienia dotyczące składni i semantyki języków programowania.</p>		wykład podający, analiza przykładów	Zaliczenie ustne, test
TP-03	<p>Omówienie podstawowych konstrukcji programistycznych, instrukcje: przypisania, warunkowe, wyboru, składnia i semantyka poszczególnych instrukcji. Przykłady programów w C i C++.</p>		wykład podający, analiza przykładów	Zaliczenie ustne, test
TP-04	<p>Instrukcje iteracyjne (pętle for, while) – schematy pętli, analiza działania na przykładach. Złożone typy danych: tablice jednowymiarowe, wielowymiarowe, znakowe. Błędy obliczeń podczas stosowania pętli.</p>		wykład podający, analiza przykładów	Zaliczenie ustne, test

TP-05	Podprogramy: funkcje. Widoczność zmiennych, sposoby przekazywania argumentów do funkcji, wartość zwracana przez funkcje. Funkcje rekurencyjne i biblioteczne. Przykłady w języku C i C++.		wykład podający, analiza przykładów	Zaliczenie ustne, test
TP-06	Pojęcie wskaźnika, dynamiczne przydzielanie i zwalnianie pamięci dla danych alokowanych na stacku. Poruszanie się po tablicy za pomocą wskaźnika. Arytmetyka wskaźników. Wykorzystanie wskaźników w funkcjach.		wykład podający, analiza przykładów	Zaliczenie ustne, test
TP-07	Typ strukturalny łączący dane. Tworzenie struktur oraz tablic struktur. Posługiwanie się składnikami struktur. Zagnieżdżanie typów strukturalnych. Operacje plikowe w programach. Otwieranie i zamykanie pliku. Zapis i odczyt sformatowany. Zapis i odczyt do/z pliku. Przykładowe programy w języku C i C++.		wykład podający, analiza przykładów	Zaliczenie ustne, test
		laboratorium		

TP-08	<p>Tworzenie, wczytywanie i zapisywanie projektów w wybranym środowisku programistycznym języka C++. Operacje edycyjne, konfiguracyjne i awaryjne. Pisanie przykładowych programów prezentujących podstawowe konstrukcje programistyczne – wprowadzanie danych z klawiatury. Programy realizujące obliczenia na liczbach naturalnych, całkowitych i zmiennoprzecinkowych, funkcje matematyczne.</p>		<p>ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy, analiza przykładów</p>	<p>kolokwium, ocena realizacji zadań praktycznych, obserwacja pracy studenta</p>
TP-09	<p>Pisanie prostych programów wykorzystujących instrukcje warunkowe if oraz switch. Zastosowanie pętli for do wyprowadzania i obliczania powtarzających się wyrażeń algorytmicznych. Zagnieżdżanie pętli. Przykładowe programy wykorzystujące instrukcje iteracyjne.</p>		<p>ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy, analiza przykładów</p>	<p>kolokwium, ocena realizacji zadań praktycznych, obserwacja pracy studenta</p>
TP-10	<p>Przykładowe programy wykorzystujące niedeterministyczne pętle typu while oraz do-while. Zastosowanie złożonego typu danych – deklaracja tablic jedno-, dwu- i wielowymiarowych w programach. Obsługa poszczególnych elementów tablicy. Pisanie programów operujących tablicami z wykorzystaniem instrukcji iteracyjnych.</p>		<p>ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy, analiza przykładów</p>	<p>kolokwium, ocena realizacji zadań praktycznych, obserwacja pracy studenta</p>

TP-11	<p>Podział programu na podprogramy - użycie funkcji. Argumenty funkcji oraz sposoby przekazywania argumentów do funkcji. Sposoby deklarowania i definiowania nowych funkcji w języku C++.</p> <p>Wykorzystanie w programach funkcji bibliotecznych, dołączanie bibliotek do programów.</p> <p>Zastosowania zmiennych lokalnych i globalnych.</p>		<p>ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy, analiza przykładów</p>	<p>kolokwium, ocena realizacji zadań praktycznych, obserwacja pracy studenta</p>
TP-12	<p>Wykorzystanie typu tablicowego i instrukcji iteracyjnych oraz funkcji w pisaniu programów operujących tablicami. Zastosowanie algorytmów sortowania, przeszukiwania i zliczania do operowania tablicami – pisanie przykładowych programów w C++.</p> <p>Łańcuchy jako przykład typu tablicowego, operacje na łańcuchach, modelowanie tekstów.</p>		<p>ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy, analiza przykładów</p>	<p>kolokwium, ocena realizacji zadań praktycznych, obserwacja pracy studenta</p>
TP-13	<p>Pisanie przykładowych programów z zastosowaniem wskaźników.</p> <p>Operowanie danymi poprzez wskaźniki – wykorzystanie adresów.</p> <p>Tworzenie tablic dynamicznych z zastosowaniem wskaźników – przykłady programów.</p> <p>Przekazywanie tablic do funkcji – przykłady programów w C++.</p> <p>Wykorzystanie operatora new oraz delete.</p>		<p>ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy, analiza przykładów</p>	<p>kolokwium, ocena realizacji zadań praktycznych, obserwacja pracy studenta</p>

TP-14	Wykorzystanie typów strukturalnych w programach – struct. Tworzenie nowych typów strukturalnych, operowanie składnikami struktur. Pisanie funkcji obsługujących typy strukturalne – przykłady w języku C++. Pisanie prostych programów realizujących podstawowe operacje na plikach – odczyt i zapis danych do plików – przykładowe programy.		ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy, analiza przykładów	kolokwium, ocena realizacji zadań praktycznych, obserwacja pracy studenta
-------	---	--	---	---

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 2008
2. Schildt H.: Programowanie C++, Wydawnictwo RM 2002
3. Eckel B.: Thinking in C++ : edycja polska, Helion 2004
4. Prata S.: Język C++: szkoła programowania, Wydawnictwo Robomatic, Wrocław 2002

Literatura uzupełniająca:

1. Kubiak M.: C++. Zadania z programowania z przykładowymi rozwiązaniami. Wydanie III, Helion 2020
2. Wróblewski P.: Algorytmy, struktury danych i techniki programowania: podstawowy podręcznik do nauki algorytmiki , Wyd. 4., Helion, Gliwice 2010
3. Stabrowski M.: Język C++ w przykładach , Wyższa Szkoła Ekonomiczno-Informatyczna w Warszawie, 2005

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	60
SUMA GODZIN:	120

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: 4	2

	Praca własna studenta		2
* 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 symboleffektu 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.			
Czytanie wskazanej literatury (M_01-M_03, M_07,M_08), przygotowanie do zajęć (M_01-M_03, M_04-M_06, M_07,M_08), przygotowanie do kolokwium (M_01-M_03, M_04-M_06, M_07,M_08), samodzielne pisanie programów (M_01-M_03, M_04-M_06, M_07,M_08),			
KRYTERIA OCENIANIA			
Ocena kształtująca: ocena przygotowania do zajęć ocena umiejętności stosowania zdobytej wiedzy podczas wykonywania zadań realizowanych podczas zajęć ocena aktywności podczas zajęć			
Ocena podsumowująca: ocena umiejętności rozwiązywania postawionych problemów ocena zadań implementacyjnych do samodzielnego wykonania zaliczenie wykładów na podstawie kolokwium zaliczenie zajęć praktycznych na podstawie kolokwium, realizacji zadań i obserwacji pracy studenta			
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 (sylabus) musi być dostępna dla studenta.