

## Karta opisu zajęć - Sylabus

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

### I. INFORMACJE PODSTAWOWE

Nazwa zajęć: Podstawy programowania	Cykl kształcenia: 2022/2023	Data aktualizacji sylabusa: 2022-10-01
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	Koordynator zajęć Imię, nazwisko, tytuł/stopień naukowy, adres e-mail: Przemysław Hawro, dr inż. przemyslaw.hawro@pwste.edu.pl	
Jednostka organizacyjna: Instytut Inżynierii Technicznej	Prowadzący zajęcia Imię, nazwisko, tytuł/stopień naukowy, adres e-mail: Przemysław Hawro, dr inż. przemyslaw.hawro@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:		Laboratorium:	
Lektorat:		Lektorat:	
Projekt:		Projekt:	
Zajęcia praktyczne:	45	Zajęcia praktyczne:	
Seminarium:		Seminarium:	
Zajęcia terenowe:		Zajęcia terenowe:	
Praktyki:		Praktyki:	
Inna forma (jaka):		Inna forma (jaka):	
<b>RAZEM:</b>	60	<b>RAZEM:</b>	

### 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.

<b>Efekty uczenia się określone dla zajęć</b>				
<p>Efekty uczenia się określone dla zajęć w kategorii wiedza, umiejętności oraz kompetencje społeczne oraz metody weryfikacji efektów uczenia się</p> <p><b>UWAGA:</b> 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.</p>				
Symbol efektów uczenia się określonego dla zajęć*		Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się, student w kategorii:		
<b>Wiedzy - zna i rozumie</b>				
M_01	Zna podstawowe metody, techniki i narzędzia stosowane przy rozwiązywaniu prostych zadań informatycznych z zakresu teorii algorytmów, budowy systemów komputerowych, języków programowania.			
M_02	Charakteryzuje podstawowe pojęcia programistyczne: zadanie algorytmiczne, selekcja, iteracja, funkcja, rekurencja. Dobiera i stosuje podstawowe definicje algorytmiczno-programistyczne.			
M_03	Dobiera metody projektowania oprogramowania zgodnie z metodyką strukturalną.			
<b>Umiejętności - potrafi</b>				
M_04	Ustala kryteria integracji uzyskanych informacji z odpowiednią ich interpretacją.			
M_05	Wybiera metodę algorytmiczną do postawionego problemu oraz odpowiednie środowisko programistyczne.			
M_06	Opracowuje poszczególne etapy projektu oraz konstruuje optymalne rozwiązanie zadanego problemu.			
<b>Kompetencji społecznych - jest gotów do</b>				
M_07	Jest otwarty na podnoszenie swoich kwalifikacji zawodowych poprzez uczestnictwo w szkoleniach specjalistycznych.			
M_08	Wykazuje odpowiedzialność za przestrzeganie zasad poszanowania nadrzędnej roli człowieka w szybko rozwijającej się technologii informacyjnej.			
<p><b>UWAGA!</b> Zaleca się, aby w zależności od liczby godzin zajęć, liczba efektów uczenia się zawierała się w przedziale: 3-7, ale są to wartości umowne.</p>				
<b>TREŚCI PROGRAMOWE I ICH ODNIESIENIE DO EFEKTÓW UCZENIA SIĘ OKREŚLONYCH DLA ZAJĘĆ</b>				
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 (symbol efektów uczenia się)
		<b>wykład</b>		

TP-01	<p>Omówienie tematyki przedmiotu, literatury, form i zasad zaliczenia.</p> <p>Pojęcie paradygmatu programowania strukturalnego. Struktura programu w języku C++, pliki źródłowe i nagłówkowe. Pojęcie algorytmu, przykłady podstawowych algorytmów w postaci schematów graficznych.</p>		2	M_01, M_02, M_03, M_04, M_05, M_06, M_07, M_08
TP-02	<p>Wybrane środowiska programistyczne. Sposoby zapisu danych liczbowych w programie. Wykorzystanie zmiennych i stałych w programach. Rozwiązywanie prostych problemów algorytmicznych – rysowanie schematów blokowych oraz implementacja w kodzie języka C++. Wykorzystanie w programach wyrażeń arytmetycznych i logicznych. Zagadnienia dotyczące składni i semantyki języków programowania.</p>		2	M_01, M_02, M_03, M_04, M_05, M_06, M_07, M_08
TP-03	<p>Omówienie podstawowych konstrukcji programistycznych, instrukcje: przypisania, warunkowe, wyboru, składnia i semantyka poszczególnych instrukcji. Przykłady programów w C i C++.</p>		2	M_01, M_02, M_03, M_04, M_05, M_06, M_07, M_08
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>		2	M_01, M_02, M_03, M_04, M_05, M_06, M_07, M_08
TP-05	<p>Podprogramy: funkcje. Widoczność zmiennych, sposoby przekazywania argumentów do funkcji, wartość zwracana przez funkcje. Funkcje rekurencyjne i biblioteczne. Przykłady w języku C i C++.</p>		2	M_01, M_02, M_03, M_04, M_05, M_06, M_07, M_08

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.		2	M_01, M_02, M_03, M_04, M_05, M_06, M_07, M_08
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++.		2	M_01, M_02, M_03, M_04, M_05, M_06, M_07, M_08
		<b>zajęcia praktyczne</b>		
TP-08	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.		5	M_01, M_02, M_03, M_04, M_05, M_06, M_07, M_08
TP-09	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.		8	M_01, M_02, M_03, M_04, M_05, M_06, M_07, M_08

TP-10	<p>Przykładowe programy wykorzystujące niedeterministyczne pętle typu while oraz do-while.</p> <p>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>		8	M_01, M_02, M_03, M_04, M_05, M_06, M_07, M_08
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. Zastosowania zmiennych lokalnych i globalnych.</p>		6	M_01, M_02, M_03, M_04, M_05, M_06, M_07, M_08
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++. Łańcuchy jako przykład typu tablicowego, operacje na łańcuchach, modelowanie tekstów.</p>		6	M_01, M_02, M_03, M_04, M_05, M_06, M_07, M_08
TP-13	<p>Pisanie przykładowych programów z zastosowaniem wskaźników. 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>		6	M_01, M_02, M_03, M_04, M_05, M_06, M_07, M_08

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.		6	M_01, M_02, M_03, M_04, M_05, M_06, M_07, M_08
-------	---	--	---	--

### 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. 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

**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 #
	<b>Wiedza</b>	wykład	
M_01	TP_01-TP_14	wykład podający, ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy	kolokwium
M_02	TP_01-TP_14	wykład podający, ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy	kolokwium
M_03	TP_01-TP_14	wykład podający, ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy	kolokwium
	<b>Umiejętności</b>	ćwiczenia, laboratorium, projekt, zajęcia praktyczne	
M_04	TP_01-TP_14	ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy	kolokwium, zaliczenie projektu praktycznego

M_05	TP_01-TP_14	ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy	kolokwium, zaliczenie projektu praktycznego
M_06	TP_01-TP_14	ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy	kolokwium, zaliczenie projektu praktycznego
	<b>Kompetencje społeczne</b>	ćwiczenia, laboratorium, projekt, zajęcia praktyczne	
M_07	TP_01-TP_14	ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy, studium przypadku	Obserwacja pracy studenta
M_08	TP_01-TP_14	ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy, studium przypadku	Obserwacja pracy studenta
<p><b>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.</b></p> <p>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.</p>			
<b>BILANS PUNKTÓW ECTS</b>			
<b>OBCIĄŻENIE PRACĄ STUDENTA (godziny)</b>			
Forma aktywności		Liczba godzin *	
Godziny zajęć (według harmonogramu) z nauczycielem akademickim lub inną osobą prowadzącą zajęcia		60	
Praca własna studenta		60	
<b>SUMA GODZIN:</b>		120	
<b>OBCIĄŻENIE PRACĄ STUDENTA (punkty ECTS)</b>			
		Liczba punktów ECTS	
SUMARYCZNA LICZBA PUNKTÓW ECTS PRZYPIŚANYCH 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;			
<b>OPIS PRACY WŁASNEJ STUDENTA:</b>			
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.			
M_01 - M_03	czytanie wskazanej literatury, przygotowanie do kolokwium	ćwiczenia praktyczne, projekt, kolokwium	

M_4 - M_06	opracowanie zadań i projektu, przygotowanie do zajęć	ćwiczenia praktyczne, projekt, kolokwium
M_7 - M_08	czytanie wskazanej literatury	ćwiczenia praktyczne, projekt
<b>KRYTERIA OCENIANIA</b>		
Ocena kształtująca:		
<ol style="list-style-type: none"> <li>1. Zrozumienie celu zajęć przez studenta.</li> <li>2. Raportowanie postępu realizacji ćwiczenia</li> <li>3. Zaangażowanie studenta w wykonywane ćwiczenie</li> <li>4. Samoocena i ocena koleżeńska.</li> </ol>		
Ocena podsumowująca:		
<p>Na ocenę dostateczną student ma podstawową wiedzę w zakresie zasad tworzenia algorytmów i technik programowania w języku C++. Student nabył umiejętność analizy i wytwarzania kodu oraz potrafi dokonać implementacji prostego problemu programistycznego w języku C++. Student potrafi samodzielnie rozwiązać proste zadania i problemy programistyczne oraz uruchomić poprawnie wykonany kod źródłowy z użyciem środowiska programistycznego.</p> <p>Na ocenę dobrą student ma zadowalającą wiedzę w zakresie zasad tworzenia algorytmów i technik programowania w języku C++. Student nabył umiejętność analizy i wytwarzania kodu oraz potrafi dokonać implementacji problemu programistycznego w języku C++. Student potrafi samodzielnie rozwiązać większość zadań i problemów programistycznych oraz uruchomić poprawnie wykonany kod źródłowy z użyciem środowiska programistycznego.</p> <p>Na ocenę bardzo dobrą student ma rozszerzoną wiedzę w zakresie zasad tworzenia algorytmów i technik programowania w języku C++. Student nabył umiejętność analizy i wytwarzania kodu oraz potrafi dokonać implementacji problemu programistycznego w języku C++. Student potrafi samodzielnie rozwiązać wszystkie zadania czy problemy programistyczne i uruchomić poprawnie wykonany kod źródłowy z użyciem środowiska programistycznego.</p>		
<b>INFORMACJA O PRZEWIDYWANEJ MOŻLIWOŚCI WYKORZYSTANIA B-LEARNINGU</b>		
istnieje		
<b>INFORMACJA O PRZEWIDYWANEJ MOŻLIWOŚCI WYKORZYSTANIA E-LEARNINGU</b>		
istnieje		

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

.....  
 (data, podpis Dyrektora Instytutu/  
 Kierownika Jednostki Międzyinstytutowej)

.....  
 (data, podpis Kierownika Zakładu)

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