

## Karta opisu zajęć - Sylabus

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

### I. INFORMACJE PODSTAWOWE

Nazwa zajęć: Podstawy programowania	Cykl kształcenia: 2022-2026	Data aktualizacji sylabusu: 2022-02-18
Nazwa kierunku studiów, poziom i profil kształcenia: Informatyka, studia pierwszego stopnia, profil praktyczny		
Język wykładowy: polski	Rodzaj zajęć: Zajęcia kształcenia kierunkowego	
Rok studiów: 1	Semestr: 1	
Liczba punktów ECTS przypisana zajęciom: 6	Koordynator zajęć Imię, nazwisko, tytuł/stopień naukowy, adres e-mail: Tomasz Lewandowski, dr inż., tomasz.lewandowski@pwste.edu.pl	
Jednostka organizacyjna: Instytut Inżynierii Technicznej	Prowadzący zajęcia Imię, nazwisko, tytuł/stopień naukowy, adres e-mail: Tomasz Lewandowski, dr inż., tomasz.lewandowski@pwste.edu.pl Monika Pirog, dr monika.pirog@pwste.edu.pl 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:	30	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):	
<b>RAZEM:</b>	75	<b>RAZEM:</b>	

### II. INFORMACJE SZCZEGÓŁOWE

#### Wymagania wstępne i dodatkowe:

Podstawowe umiejętności matematyczne oraz informatyczne na poziomie szkoły ponadpodstawowej

<b>Cel (cele) kształcenia dla zajęć:</b>				
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>				
Efekty uczenia się określone dla zajęć w kategorii wiedza, umiejętności oraz kompetencje społeczne oraz metody weryfikacji efektów uczenia się				
<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.				
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	podstawowe metody, techniki i narzędzia stosowane w programowaniu imperatywnym			
M_02	podstawowe pojęcia programistyczne: zadanie algorytmiczne, instrukcja warunkowa, iteracja, rekurencja; budowę programu w języku C++, zakresów dostępności zmiennych, budowę funkcji i sposoby przekazywania parametrów, wykorzystywanie i rola wskaźników, struktur i unii w języku C++			
<b>Umiejętności - potrafi</b>				
M_03	wybrać odpowiednią metodę algorytmiczną do postawionego problemu oraz napisać program w języku C++ realizujący określone wymagania funkcjonalne			
M_04	korzystać z gotowych bibliotek dostarczanych wraz z językiem programowania w projektowaniu i implementacji oprogramowania			
<b>Kompetencji społecznych - jest gotów do</b>				
M_05	pozyskiwania informacji z zasobów internetowych oraz literatury niezbędnych do rozwiązania napotkanych problemów			
M_06	podnoszenia swoich kwalifikacji zawodowych poprzez uczestnictwo w szkoleniach specjalistycznych oraz zdobywanie certyfikatów developerskich			
<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.				
<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	Współczesne aspekty programowania. Pojęcie języka strukturalnego, struktura programu w języku C++, pliki źródłowe i nagłówkowe. Pojęcie algorytmu, przykłady podstawowych algorytmów.		6	M_01, M_02, M_03, M_04, M_05, M_06
TP-02	Wybrane środowiska programistyczne. Strumienie wejścia/wyjścia. Typy i formaty zmiennych, konwersja typów.		2	M_01, M_02, M_03, M_04, M_05, M_06
TP-03	Podstawowe konstrukcje programistyczne, instrukcje: przypisania, warunkowa, wyboru. Operatory arytmetyczne, relacyjne, logiczne, priorytety operatorów.		2	M_01, M_02, M_03, M_04, M_05, M_06
TP-04	Instrukcje iteracyjne for, while, do-while. Złożone typy danych: tablice jednowymiarowe, wielowymiarowe, znakowe, typy wyliczeniowe.		4	M_01, M_02, M_03, M_04, M_05, M_06
TP-05	Definicja a deklaracja funkcji. Widoczność zmiennych, sposoby przekazywania argumentów do funkcji, wartość zwracana przez funkcje. Funkcje rekurencyjne i biblioteczne.		2	M_01, M_02, M_03, M_04, M_05, M_06
TP-06	Pojęcie wskaźnika, dynamiczne alokowanie i zwolnienie pamięci dla tablic. Poruszanie się po tablicy za pomocą wskaźnika. Arytmetyka wskaźników. Wykorzystanie wskaźników w funkcjach.		4	M_01, M_02, M_03, M_04, M_05, M_06
TP-07	Statyczne wydzielenie pamięci i inicjalizacja. Dynamiczne alokowanie i zwolnienie pamięci.		2	M_01, M_02, M_03, M_04, M_05, M_06
TP-08	Typ strukturalny łączący dane. Tworzenie struktur oraz tablic struktur. Posługiwanie się składnikami struktur. Zagnieżdżanie struktur.		4	M_01, M_02, M_03, M_04, M_05, M_06
TP-09	Dynamiczne struktury danych: kolejka, stos, lista. Otwieranie i zamykanie pliku. Zapis i odczyt sformatowany. Zapis i odczyt do/z pliku.		4	M_01, M_02, M_03, M_04, M_05, M_06
		<b>laboratorium</b>		

TP-10	Tworzenie, wczytywanie i zapisywanie projektów. Operacje edycyjne, konfiguracyjne i awaryjne. Wprowadzenie danych z klawiatury i wyprowadzenie na monitor. Operacje arytmetyczne, priorytet operacji, konwersja typów.		5	M_01, M_02, M_03, M_04, M_05, M_06
TP-11	Pisanie prostych programów, programy z rozgałęzieniami z zastosowaniem instrukcji warunkowych, programy iteracyjne wykorzystujące niedeterministyczne pętle typu while oraz do-while.		4	M_01, M_02, M_03, M_04, M_05, M_06
TP-12	Zastosowanie pętli for do wyprowadzania i obliczania powtarzających się wyrażeń. Tablice jedno, dwu i wielowymiarowe. Tablice dynamiczne.		6	M_01, M_02, M_03, M_04, M_05, M_06
TP-13	Podział programu na podprogramy, użycie funkcji. Funkcje obsługi zdarzeń, parametry przekazywane przez referencję i przez wartość. Deklaracja i definicja podprogramów, funkcje wbudowane i definiowane. Zmienne lokalne, zmienne globalne. Przekazywanie argumentów do funkcji.		6	M_01, M_02, M_03, M_04, M_05, M_06
TP-14	Instrukcje iteracyjne w algorytmach sortowania, przeszukiwania i zliczania. Łącuchy jako przykład typu tablicowego, operacje na łańcuchach, modelowanie tekstów.		6	M_01, M_02, M_03, M_04, M_05, M_06
TP-15	Zastosowanie wskaźników, adres i zmienna. Operator referencji i dereferencji. Operowanie danymi poprzez wskaźniki. Stos i sarta. Operator new i delete.		6	M_01, M_02, M_03, M_04, M_05, M_06
TP-16	Przykłady wykorzystujące strukturalny jednowymiarowy typ danych – struct – do gromadzenia i przetwarzania bloków danych różnego typu. Standardowe wejście i standardowe wyjście - dostęp do plików. Zapisywanie i odczytywanie struktur danych.		6	M_01, M_02, M_03, M_04, M_05, M_06

TP-17	Wykonanie projektu aplikacji wykorzystującej: strumienie wejścia/wyjścia, różne typy i formaty zmiennych, złożone typy danych, konstrukcje programistyczne, operatory, funkcje oraz inne możliwości i techniki poznane w trakcie wykładu i ćwiczeń laboratoryjnych		5	M_01, M_02, M_03, M_04, M_05, M_06
-------	--	--	---	--

**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_17	wykład podający, ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy	egzamin, kolokwium
M_02	TP_01-TP_17	wykład podający, ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy	egzamin, kolokwium
	<b>Umiejętności</b>	laboratorium	
M_03	TP_01-TP_17	ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy	kolokwium, zaliczenie projektu praktycznego
M_04	TP_01-TP_17	ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy	kolokwium, zaliczenie projektu praktycznego
	<b>Kompetencje społeczne</b>	laboratorium	

M_05	TP_01-TP_17	ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy, studium przypadku,	Obserwacja pracy studenta
M_06	TP_01-TP_17	ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy, studium przypadku,	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.

### 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
<b>SUMA GODZIN:</b>	<b>150</b>

#### 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.

M01, M_02	czytanie wskazanej literatury, przygotowanie do egzaminu	ćwiczenia praktyczne, projekt, egzamin
M_03	opracowanie zadania, projektu	ćwiczenia praktyczne, projekt
M_04	przygotowanie do zajęć	ćwiczenia praktyczne, projekt
M_05	czytanie wskazanej literatury	ćwiczenia praktyczne, projekt

### KRYTERIA OCENIANIA

Ocena kształtująca:

1. Zrozumienie celu zajęć przez studenta.
2. Raportowanie postępu realizacji ćwiczenia
3. Zaangażowanie studenta w wykonywane ćwiczenie
4. Samoocena i ocena koleżeńska.

Ocena podsumowująca:

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.

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.

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.

**INFORMACJA O PRZEWDYWANEJ MOŻLIWOŚCI WYKORZYSTANIA B-LEARNINGU**

istnieje

**INFORMACJA O PRZEWDYWANEJ MOŻLIWOŚCI WYKORZYSTANIA E-LEARNINGU**

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.*