

## KARTA PRZEDMIOTU

Nazwa w języku polskim: **Języki programowania**

Nazwa w języku angielskim: **Programming languages**

Kierunek studiów (jeśli dotyczy): **Biomechanika Inżynierska**

Specjalność (jeśli dotyczy):

Poziom i forma studiów: **II stopień, stacjonarne**

Rodzaj przedmiotu: **wybieralny**

Kod przedmiotu: **BIM041022**

Grupa kursów: **nie**

	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium
Liczba godzin zajęć zorganizowanych w Uczelni (ZZU)	30			15	
Liczba godzin całkowitego nakładu pracy studenta (CNPS)	60			30	
Forma zaliczenia	Zaliczenie na ocenę			Zaliczenie na ocenę	
Grupa kursów					
Liczba punktów ECTS	2			1	
w tym liczba punktów odpowiadająca zajęciom o charakterze praktycznym (P)				1	
w tym liczba punktów ECTS odpowiadająca zajęciom wymagającym bezpośredniego kontaktu (BK)	1.2			0.7	

### WYMAGANIA WSTĘPNE W ZAKRESIE WIEDZY, UMIEJĘTNOŚCI I KOMPETENCJI SPOŁECZNYCH

1. Zakłada się, że przed rozpoczęciem nauki niniejszego przedmiotu student posiada przygotowanie w zakresie: systemów liczenia, prostych algorytmów, podstawowych elementów programowania w języku C i podstawowej wiedzy o budowie i działaniu komputerów.

### CELE PRZEDMIOTU

- C1. Zapoznanie słuchaczy z podstawowymi technikami programowania strukturalnego.
- C2. Zapoznanie słuchaczy ze sposobami komputerowej reprezentacji danych.
- C3. Przygotowanie do samodzielnego tworzenia oprogramowania i realizacji prostych algorytmów.

## PRZEDMIOTOWE EFEKTY UCZENIA SIĘ

### I. Z zakresu wiedzy:

PEK\_W01 - W wyniku przeprowadzonych zajęć student ma podstawową wiedzę dotyczącą tworzenia programów strukturalnych i obiektowych.

PEK\_W02 - W wyniku przeprowadzonych zajęć student ma podstawową wiedzę z zakresu typu zmiennych, definiowania funkcji, klas i przekazywania do nich wybranych parametrów, przeładowania operatorów.

PEK\_W03 - W wyniku przeprowadzonych zajęć student ma podstawową wiedzę z zakresu tworzenia instrukcji warunkowych, pętli programowych, operacji na wskaźnikach i tworzeniu obiektów.

### II. Z zakresu umiejętności:

PEK\_U01 - Potrafi napisać dowolny program w języku C/C++.

PEK\_U02 - Potrafi tworzyć dynamiczne struktury danych.

PEK\_U03 - Potrafi zaimplementować opracowany algorytm.

### III. Z zakresu kompetencji społecznych:

PEK\_K01 - Potrafi pracować nad zadaniami samodzielnie i w grupie.

PEK\_K02 - Potrafi myśleć i działać kreatywnie.

## TREŚCI PROGRAMOWE

Forma zajęć – Wykład		Liczba godzin
Wy1	Wprowadzenie podstawowych pojęć: algorytm i sposoby jego przedstawiania, przykłady algorytmów.	2
Wy2	Wprowadzenie do języka C. Typy danych. Działania na typach danych.	2
Wy3	Operatory: argumentowość, priorytet. Instrukcje sterujące i pętle.	2
Wy4	Działania na wskaźnikach, tablicach i funkcjach. Definicje funkcji, prototypu funkcji, wywołanie funkcji.	2
Wy5	Wartość zwracana przez funkcję. Rekurencja.	2
Wy6	Standardowe operacje wejścia i wyjścia dla języka C (stdio.h).	2
Wy7	Struktura, unia i pola bitowe: deklaracje i implementacja. Parametry funkcji main.	2
Wy8	Wprowadzenie do programowania zorientowanego obiektowo: język C++.	2
Wy9	Pojęcie klasy. Działanie na obiektach, funkcje składowe: deklarowanie i definiowanie. Wskaźnik this. Składnik statyczny klasy.	2
Wy10	Kapsułkowanie (enkapsulacja), różnica między strukturą a klasą w języku C++.	2
Wy11	Przesłanianie nazw zmiennych i funkcji. Przeładowanie nazw funkcji. Argumenty domyślne funkcji. Wprowadzenie do konstruktora.	2
Wy12	Konstruktor. Destruktor. Dynamiczna alokacja pamięci.	2
Wy13	Lista inicjalizacyjna konstruktora. Konstruktor kopiujący.	2
Wy14	Funkcje zaprzyjaźnione. Zaprzyjaźnienie klas.	2
Wy15	Przeładowanie operatorów: liczba argumentów; operator jako funkcja zwykła, jako funkcja składowa.	2

		Suma: 30
Forma zajęć – Laboratorium		Liczba godzin
Lab1	Zajęcia wprowadzające. Zapoznanie się ze programistycznym środowiskiem pracy.	1
Lab2	Tworzenie projektu i realizacja prostych algorytmów. Zastosowanie instrukcji sterujących i pętli.	2
Lab3	Deklarowanie i definiowanie funkcji.	2
Lab4	Tworzenie tablic, struktur i unii. Alokacja dynamiczna.	2
Lab5	Działanie na wskaźnikach i tablicach.	2
Lab6	Operacje logiczne i arytmetyczne (bitowe).	2
Lab7	Projekt własny	2
Lab8	Projekt własny	2
		Suma: 15
Forma zajęć – Projekt		Liczba godzin
Proj1	Zajęcia wprowadzające. Zapoznanie się ze programistycznym środowiskiem pracy.	1
Proj2	Tworzenie projektu i realizacja prostych algorytmów. Zastosowanie instrukcji sterujących i pętli.	2
Proj3	Deklarowanie i definiowanie funkcji.	2
Proj4	Tworzenie tablic, struktur i unii. Alokacja dynamiczna.	2
Proj5	Działanie na wskaźnikach i tablicach.	2
Proj6	Operacje logiczne i arytmetyczne (bitowe).	2
Proj7	Projekt własny	2
Proj8	Projekt własny	2
		Suma: 15

STOSOWANE NARZĘDZIA DYDAKTYCZNE	
N1. wykład informacyjny N2. prezentacja multimedialna N3. przygotowanie sprawozdania	

OCENA OSIĄGNIĘCIA PRZEDMIOTOWYCH EFEKTÓW UCZENIA SIĘ (Wykład)		
Oceny (F – formująca (w trakcie semestru), P – podsumowująca (na koniec semestru))	Numer efektu uczenia się	Sposób oceny osiągnięcia efektu uczenia się
F1	PEK_W01, PEK_W02, PEK_W03	kolokwium
P = F1		

OCENA OSIĄGNIĘCIA PRZEDMIOTOWYCH EFEKTÓW UCZENIA SIĘ (Projekt)		
Oceny (F – formująca (w trakcie semestru), P – podsumowująca (na koniec semestru))	Numer efektu uczenia się	Sposób oceny osiągnięcia efektu uczenia się
F1	PEK_U01, PEK_U02, PEK_U03, PEK_K01, PEK_K02	raport
P = F1		

LITERATURA PODSTAWOWA I UZUPEŁNIAJĄCA	
<p><u>LITERATURA PODSTAWOWA</u>  [1] Grębosz Jerzy, Symfonia C++, Oficyna Kallimach, Kraków 1993,  [2] Stroustrup Bjarne, Język C++, WNT, Warszawa 1994.</p> <p><u>LITERATURA UZUPEŁNIAJĄCA</u>  [1] Grębosz J.: Pasja C++. Oficyna Kallimach, 1997.</p>	

OPIEKUN PRZEDMIOTU
dr inż. Ewelina Świątek-Najwer tel.: 71 320-21-93 email: ewelina.swiatek@pwr.edu.pl

## SUBJECT CARD

Name in Polish: **Języki programowania**

Name in English: **Programming languages**

Main field of study (if applicable): **Engineering Biomechanics**

Specialization (if applicable):

Level and form of studies: **II level, full-time**

Kind of subject: **optional**

Subject code: **BIM041022**

Group of courses: **no**

	Lecture	Classes	Laboratory	Project	Seminar
Number of hours of organized classes in University (ZZU)	30			15	
Number of hours of total student workload (CNPS)	60			30	
Form of crediting	Crediting with grade			Crediting with grade	
Group of courses					
Number of ECTS points	2			1	
including number of ECTS points for practical (P) classes				1	
including number of ECTS points for direct teacher-student contact (BK) classes	1.2			0.7	

### PREREQUISITES RELATING TO KNOWLEDGE, SKILLS AND OTHER COMPETENCES

1. It is assumed that prior to learning of this course the student has to prepare in the following areas: systems of counting, simple algorithms, the basic elements of programming in C language and basic knowledge of the design and operation of computers.

### SUBJECT OBJECTIVES

- C1. Familiarize students with the basic techniques of structured programming.
- C2. Familiarize students with the methods of computer data representation.
- C3. Prepared for independent software development and implementation of simple algorithms.

## SUBJECT LEARNING OUTCOMES

### I. Relating to knowledge:

PEK\_W01 - As a result of the course the student has a basic knowledge of structural programming and object-oriented.

PEK\_W02 - As a result of the course the student has a basic knowledge of the type of variables, defining functions, classes, and transfer them to your selection, overloading operators.

PEK\_W03 - As a result of the course the student has a basic knowledge of the creation of conditional statements, loops programming, operations on indicators and creating objects.

### II. Relating to skills:

PEK\_U01 - Can write any program in C / C++.

PEK\_U02 - Can create dynamic data structures.

PEK\_U03 - Can implement the developed algorithm.

### III. Relating to social competences:

PEK\_K01 - Can work on tasks independently and in groups.

PEK\_K02 - Can think and act creatively.

## PROGRAM CONTENT

Form of classes – Lecture		Number of hours
Lec1	The introduction of the basic concepts of: algorithm and methods of its presentation, examples of algorithms.	2
Lec2	Introduction to the C language. Data types. Working on data types.	2
Lec3	Operators: arguments, priority. Control statements and loops.	2
Lec4	Working with indicators, tables and functions. Definitions of functions, a function prototype, a function call.	2
Lec5	The value returned by the function. Recursion.	2
Lec6	The standard input and output operations for C (stdio.h).	2
Lec7	Structure, union and bit fields: the declarations and implementation. The parameters of the main function.	2
Lec8	Introduction to object-oriented programming: C++ language.	2
Lec9	The concept of class. Action on objects, member functions: declaring and defining. This indicator. The static component class.	2
Lec10	Encapsulation, the difference between the structure and class in C++.	2
Lec11	Overriding the names of variables and functions. Overloading function names. Default arguments in function. Introduction to the constructor.	2
Lec12	The constructor. The destructor. Dynamic memory allocation.	2
Lec13	Constructor initialization list. Copy constructor.	2
Lec14	Friend Functions. Friending classes.	2

Lec15	Overloading operators: number of arguments, the operator as a standard feature, as a function of the component.	2
		Total hours: 30
Form of classes – Laboratory		Number of hours
Lab1		1
Lab2		2
Lab3		2
Lab4		2
Lab5		2
Lab6		2
Lab7		2
Lab8		2
		Total hours: 15
Form of classes – Project		Number of hours
Proj1	Initiation classes. Getting familiar with the programming work environment.	1
Proj2	Creation and implementation of a simple algorithm. The use of control statements and loops.	2
Proj3	Declaring and defining a function.	2
Proj4	Creating arrays, structures and unions. Dynamic Allocation.	2
Proj5	Working on indicators and tables.	2
Proj6	Logical and arithmetic (bit) operations .	2
Proj7	The project your own	2
Proj8	The project your own	2
		Total hours: 15

TEACHING TOOLS USED	
N1. informative lecture N2. multimedia presentation N3. report preparation	

EVALUATION OF SUBJECT LEARNING OUTCOMES ACHIEVEMENT (Lecture)		
Evaluation (F – forming (during semester), P – concluding (at semester end))	Learning outcomes number	Way of evaluating learning outcomes achievement
F1	PEK_W01, PEK_W02, PEK_W03	test
P = F1		

EVALUATION OF SUBJECT LEARNING OUTCOMES ACHIEVEMENT (Project)		
Evaluation (F – forming (during semester), P – concluding (at semester end))	Learning outcomes number	Way of evaluating learning outcomes achievement
F1	PEK_U01, PEK_U02, PEK_U03, PEK_K01, PEK_K02	report
P = F1		

PRIMARY AND SECONDARY LITERATURE	
<u>PRIMARY LITERATURE</u> [1] Grębosz Jerzy, Phymfony C++, Oficyna Kallimach, Kraków 1993, [2] Stroustrup Bjarne, C++ language, WNT, Warszawa 1994.	
<u>SECONDARY LITERATURE</u> [1] Grębosz J.: C++ passion. Oficyna Kallimach, 1997.	

SUBJECT SUPERVISOR	
dr inż. Ewelina Świątek-Najwer tel.: 71 320-21-93 email: ewelina.swiatek@pwr.edu.pl	