

FACULTY OF ELECTRICAL
ENGINEERING**SUBJECT CARD**

Name in Polish: **Programowanie w języku C**
 Name in English: **Programming in the C language**
 Main field of study (if applicable): **Electrical Engineering**
 Specialization (if applicable):
 Level and form of studies: **1st level, full-time**
 Kind of subject: **obligatory**
 Subject code: **ELR052502**
 Group of courses: **NO**

	Lecture	Classes	Laboratory	Project	Seminar
Number of hours of organized classes in University (ZZU):	30		30		
Number of hours of total student workload (CNPS):	60		60		
Form of crediting:	crediting with grade		crediting with grade		
For group of courses mark (X) final course:					
Number of ECTS points:	2		2		
including number of ECTS points for practical (P) classes :			2		
including number of ECTS points for direct teacher-student contact (BK) classes:	1.40		1.40		

PREREQUISITES RELATING TO KNOWLEDGE, SKILLS AND OTHER COMPETENCES

1. Knowledge of basic problems of computer science (Information technology).
2. Abilities of handling computer with the operating system WINDOWS.

SUBJECT OBJECTIVES

- C1. Knowing and acquiring proficiency in use of principles of structural approach to creating algorithms.
 C2. Knowing of principles of programming in the C language.
 C3. Mastering a skill of writing programs in the C language.

SUBJECT LEARNING OUTCOMES*relating to knowledge:*

- PEU_W01 The student has knowledge in the scope of structural programming.
 PEU_W02 The student knows fundamentals of the C programming language.

relating to skills:

- PEU_U01 The student is able to make use of principles of structural programming.
 PEU_U02 The student is able to write a simple program in the C programming language.

relating to social competences:

- PEU_K01 The student can independently develop algorithms and simple programs in the C.

PROGRAMME CONTENT		
Form of classes - lecture		Number of hours:
Lec 1	An introduction. Algorithm. Flow charts. Idea of the structural programming.	2
Lec 2	A structure of a program in C. Identifier, data types (fundamental data types: integer, real, character, logical), declaration and initialisation of variables, defining constants. Communication through console.	2
Lec 3	Arithmetical, logical, increment, decrement and assignment operators. Calculating expressions.	2
Lec 4	Flow control in a program: bifurcation of control and jumps, loops, nested loops. A single instruction, a block of instructions; the conditional structures, conditional expressions.	2
Lec 5	The repetitive structures. continuation instruction. break instruction. The selective structure. Test.	2
Lec 6	Preprocessor: directives, macrodefinition.	2
Lec 7	A summary of utilization of instructions in the C language. Test.	2
Lec 8	Functions: the structure of a function, arguments of a function, data returned by the function, global definitions and declarations, arguments of the main function, recursivity.	2
Lec 9	Arrays (simple and multidimensional arrays), strings.	2
Lec 10	Pointers. Dynamic memory allocation.	2
Lec 11	Compound data types: enumerated types, data structures, unions. Initializing structures, and unions.	2
Lec 12	Files: opening and closing, reading and writing data.	2
Lec 13	String operations. Formatting and binary input/output.	2
Lec 14	A summary of utilization of data structures in the C language. Test	2
Lec 15	A summary of rules of developing algorithms and programs.	2
Total hours:		30

Form of classes - laboratory		Number of hours:
Lab 1	Introduction. Flow charts for simple algorithms.	2
Lab 2	Flow charts for more complex algorithms.	2
Lab 3	Writing, compiling and running simple programs - writing numbers and strings in the screen.	2
Lab 4	Writing programs with use of bifurcation of control, jumps.	2
Lab 5	Writing programs with use of loops.	2
Lab 6	Writing more complex programs with use of bifurcation of control, jumps and loops	2
Lab 7	Utilization of directives and macrodefinitions.	2
Lab 8	Programming with use of functions.	2
Lab 9	Recursive function.	2
Lab 10	Programming table operations.	2
Lab 11	Introduction pointers into program.	2
Lab 12	Writing programs with use of data structures and unions.	2
Lab 13	Creating programs which read input data from files and write results into files.	2
Lab 14	Operation on strings. Formatting in input/output operations.	2
Lab 15	Writing programs with use of different elements of the programming language.	2
Total hours:		30

TEACHING TOOLS USED
N1. Multimedia presentation. N2. Information lecture. N3. Preparation in the form of reports. N4. The C programming environment.

EVALUATION OF SUBJECT LEARNING OUTCOMES ACHIEVEMENT		
Evaluation <i>F – forming (during semester)</i> <i>P – concluding (at semester end)</i>	Educational effect number	Way of evaluating educational effect achievement
F1(W)	PEU_W01 PEU_W02	activity at the classes
F2(W)	PEU_W01 PEU_W02	average of the grades from tests
P(W)	0.1F1 + 0.9F2	
F1(L)	PEU_U01 PEU_U02	activity at the classes
F2(L)	PEU_U01 PEU_U02	reports from the classes
P(L)	0.3 F1 + 0.7 F2	

PRIMARY AND SECONDARY LITERATURE
PRIMARY LITERATURE: [1] Wirth N., Algorytmy + struktury danych = programy, WNT, Warszawa 2004. [2] Dasgupta S., Papadimitriou Ch., Vazirani U., Algorytmy, PWN, Warszawa, 2019. [3] Crawford T., Prinz P., Język C w pigułce. Kompletny przewodnik, Promise, Warszawa, 2016. [4] Kernighan B. W., Ritchie D. M., Język ANSI C. Programowanie, Helion, Gliwice 2010. [5] Prata S., Język C. Szkoła programowania. Helion, Gliwice 2016. [6] Wojtuszkiewicz K., Programowanie strukturalne i obiektowe Tom 1, PWN, Warszawa 2010..
SECONDARY LITERATURE: [1] Banachowski L., Diks K.M., Rytter W., Algorytmy i struktury danych, PWN, Warszawa 2019. [2] Kochan S.G., Język C. Kompendium wiedzy. Helion, Gliwice 2015. [3] Perry G., Miller D., Język C. Programowanie dla początkujących, Helion, Gliwice 2016. [4] Shaw Z.A., Programowanie w C. Sprytnie podejście do trudnych zagadnień, których wolałbyś unikać (takich jak język C), Helion, Gliwice 2016. [5] Tłuczek , Programowanie w języku C. Ćwiczenia praktyczne, Helion, Gliwice 2011.

SUBJECT SUPERVISOR
Kazimierz Wilkosz, kazimierz.wilkosz@pwr.edu.pl