

## DESCRIPTION OF THE COURSES

- Course code: **ELR2524**
- Course title: **Programming in C**
- Language of the lecturer: **Polish, English**

<i>Course form</i>	<i>Lecture</i>	<i>Classes</i>	<i>Laboratory</i>	<i>Project</i>	<i>Seminar</i>
<i>Number of hours/week*</i>	<b>2</b>		<b>2</b>		
<i>Number of hours/semester*</i>	<b>30</b>		<b>30</b>		
<i>Form of the course completion</i>	<b>Test</b>		<b>Performed laboratory assignments</b>		
<i>ECTS credits</i>	<b>3</b>		<b>2</b>		
<i>Total Student's Workload</i>	<b>90</b>		<b>60</b>		

- Level of the course (basic/advanced): **basic**
- Prerequisites: **credit for a course introducing into computer problems**
- Name, first name and degree of the lecturer/supervisor:  
**Wilkosz Kazimierz, PhD, DSc./Professor**
- Names, first names and degrees of the team's members:  
**dr. Kazimierz Herlender,**  
**dr. Anna Kisiel,**  
**dr. Mieczysław Kozak**  
**dr. Robert Lis,**  
**dr. Mirosław Łabuzek,**  
**dr. Robert Łukomski,**  
**dr inż. Jerzy Piotrowicz,**  
**dr. Jarosław Szymańda,**  
**dr. Leszek Woźny,**  
**dr. Ryszard Zacirka.**
- Year:.1 Semester:.1
- Type of the course (obligatory/optional): **obligatory**
- Aims of the course (effects of the course):
  - **competence in utilisation of principles of structural approach to creating algorithms,**
  - **ability of writing programs in C,**
  - **familiarising with principal ideas of object programming with use of C++ language,**
  - **enhancing practical skills in utilisation of computers.**
- Form of the teaching (traditional/e-learning): **traditional**

- Course description:  
The C Integrated Development Environment. A structure of a program in the C and C++ languages. Data types. Constants. Variables. Input/output operations. Operators. Expressions. Control Structures. Functions. Pointers. Tables. Data structures. Files. String operations. Comparison of the C language and the C++ language. Introduction into object programming: objects, classes, class constructors, class destructors, inheritance. Basics of creating algorithms.
- Lecture:

<i>Particular lectures contents</i>	<i>Number of hours</i>
1. An introduction. Algorithm. Flow charts. Idea of the structural programming.	2
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. Arithmetical, logical, increment, decrement and assignment operators. Calculating expressions.	2
3. 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, the selective structure; the repetitive structures.	2
4. Preprocessor: directives, macrodefinition.	
5. 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 2
6. Arrays (simple and multidimensional arrays), strings.	
7. Pointers. Dynamic memory allocation.	2
8. Test.	2
9. Data structures, unions: declaration of structures, declaration of structure variable, table of structures, pointers and data structures.	2 2
10. Files: opening and closing, reading and writing data.	
11. String operations. Formatting and binary input/output.	2
12. Idea of object programming. A structure of a program in C++. Objects. Classes.	2 2
13. Constructors. Destructors.	
14. Relationships between classes: friendship. Inheritance.	2
15. Final test.	2 2

- Classes – the contents:
- Seminars – the contents:
- Laboratory – the contents:
  1. Introduction. Flow charts for simple algorithms.
  2. Flow charts for more complex algorithms.
  3. Writing, compiling and running simple programs – writing numbers and strings in the screen.
  4. Writing programs with use of bifurcation of control, jumps and loops.
  5. Utilization of directives and macrodefinitions.
  6. Programming with use of functions.

7. Programming table operations.
  8. Introduction pointers into program.
  9. Writing programs with use of data structures and unions.
  10. Creating programs which read input data from files and write results into files.
  11. Programming with use of objects and classes.
  12. Utilization of constructors, destructors.
  13. Utilization of classes: friendship and inheritance.
- Project – the contents:
  - Basic literature:
    1. Wirth N., Algorytmy + struktury danych = programy. Warszawa, WNT, 2001.
    2. Kernighan B. W., Ritchie D. M., Język ANSI C, Warszawa, WNT, 2003.
    3. Sexton C., Język C to proste, Warszawa, Wyd. RM, 2001.
    4. Grębosz J., Symfonia C++. Kallimach, Kraków 2000.
  - Additional literature:
    1. Kubiak M. J., Programuję w językach Turbo Pascal i C/C++: programowanie strukturalne z elementami programowania obiektowego, Warszawa, Mikom, 2001.
    2. Stec K., Wybrane elementy języka C, Gliwice, Wyd. Pol. Śląskiej, 2001.
    3. Sexton C., Programowanie w C++ - to proste. RM, Warszawa 2001.
  - Conditions of the course acceptance/creditation:  
**Lectures: positive assessment of tests and individual assignments**  
**Laboratory: positive evaluation of all performed assignments/exercises.**

\* - depending on a system of studies