

Khmelnyskyi National University

Faculty of Information Technologies

Department of Software Engineering



APPROVE

Dean of the Faculty of IT

Hovorushchenko T.O.

09 2024

SYLLABUS

Educational discipline Programming

Educational and professional program Software engineering

First level of higher education (bachelor)

General information

Item	Content
Instructor(s)	Valeriy Martynyuk
Instructor's profile	http://ki.khnu.km.ua/team/martynyuk-valerij-volodymyrovych/
Instructor's E-mail	martynyukvv@khnu.km.ua
Phone number	By agreement
Course web page at the university site	https://msn.khmn.edu.ua/enrol/index.php?id=8709
Academic year	2024-2025
Consultations	Offline: Wednesday, 5 th class, 4-325; Friday, 5 th class, 4-325; online: by prior arrangement

Course characteristics

Course status	Study mode	Year	Semester	Total number		Number of hours							Type of semester control		
				ECTS credits	Hours	Classwork hours				Student' s individual work	Student' s independent work including individual work	Course project	Course paper	Pass/ fail test	Examination
						Total	Lectures	Laboratory works	Practical classes						
E	F	1	1	10	300	136	34	68	34		164			+	
Разом				10	300	136	34	68	34		164			1	

Course Description

In the course students plan to master the basics of programming, definition to solve the problem of programming technology, language, programming system, instrumental environment, implementation of functional and object decomposition of the program according to the chosen programming technology, program code development, correction of syntactic and semantic errors and code refactoring, program debugging and testing, programming of dynamic data structures, exception handling.

The course is taught to full-time students majoring in information technology. Active and creative forms of conducting classes are used in teaching the discipline. The task of the course involves providing students with knowledge about algorithmization and its use to solve practical problems, learn to program in C/C++.

Prerequisites: Higher mathematics; Discrete mathematics;

Co-requisites: Object-oriented programming, Data structures and algorithms

Course aim and tasks

The aim of the course "Programming" is: 1) formation of competencies necessary for abstract thinking, analysis and synthesis in the implementation of algorithmization and programming; 2) development of students' professional style of thinking on algorithmization; 3) providing knowledge of the theory of algorithms and constructions of programming languages (on the example of C), necessary for further study of special disciplines and for practical engineering activities; 4) developing students' ability to use the acquired knowledge in program development.

Tasks of discipline. To provide students with knowledge about algorithmization and its use for solving practical problems, to teach programming in C/C++.

Expected learning outcomes

A student who has successfully completed the discipline must master the basics of algorithmization and programming, compose algorithms and program them using C/C++, programming paradigms, fundamental data structures, structural programming, programming languages, recursion, programming dynamic structures data, algorithms and data structures; processing standard and non-standard exceptions, modeling, designing and developing algorithms, assessing their complexity; plan and effectively organize the work of compiling programs in the C language.

Thematic and calendar plan of studying the course

Autumn semester

№ of week	Lecture topic*	Practical class topic*	Laboratory work topic*	Student's independent work		
				Content	Hours	Literature
1	Introduction. Information Technology Introduction. Information and its presentation. Binary arithmetic. Lit. [15, pp.14-40]	Linear algorithms.	Linear algorithms. Implementation in the C language.	Elaboration of lecture material. Preparation for laboratory work №1 and practical work №1. Independent work on program development for laboratory work №1.	9	[15, C.14-40]
2	Introduction. Computer systems The general structure of the computer. Computer systems and their components. Application software. System software. Programming paradigms. Lit. [15, pp.14-40]	Linear algorithms.	Linear algorithms. Implementation in the C language.	Elaboration of lecture material. Preparation for laboratory work №1 and practical work №1. Independent work on program development for laboratory work №1.	9	[15, C.14-40]
3	Algorithms and their properties. Basic concepts Algorithms and algorithmization.	Branched algorithms.		Elaboration of lecture material. Preparation for laboratory work №2 and practical work №2.	9	[3, C.12-27]

	<p>The concept of algorithm. Algorithms and their properties. The concept of algorithms and their properties. Algorithm properties. Step-by-step detailing method. Forms of algorithms. Lit. [3, pp.12-27]</p>			<p>Independent work on program development for laboratory work №2.</p>		
4	<p>Algorithms and their properties. Types of algorithms Recording of the algorithm in the form of block diagrams. Examples. Basic algorithmic structures. Following. Branching. Repetition. Examples. Types of basic structures of algorithms. The concept of the program. Lit. [3, pp.12-27]</p>	<p>Branched algorithms.</p>		<p>Elaboration of lecture material. Preparation for laboratory work №2 and practical work №2. Independent work on program development for laboratory work №2.</p>	9	[3, C.12-27]
5	<p>Introduction to C Introduction to the C programming language. Alphabet and language dictionary. Identifiers. Basic data types. Type modifiers. Lit. [1, pp.7-18; 2, pp. 6-12; 5-14]</p>	<p>Programming of loop algorithms.</p>		<p>Elaboration of lecture material. Preparation for laboratory work №3 and practical work №3. Independent work on program development for laboratory work №3.</p>	9	[1, C.7-18; 2, C. 6-12; 5-14]
6	<p>The structure of the language program C Constants. Prefix and suffix forms. Expressions and operations. Decimal, octal</p>	<p>Programming of loop algorithms.</p>	<p>Linear algorithms. Implementation in the C language.</p>	<p>Elaboration of lecture material. Preparation for laboratory work №3 and practical work №3. Independent work on program development for laboratory work</p>	9	[1, C.7-18; 2, C. 13-20; 5-14]

	and hexadecimal representation. Identifiers. Keywords. Comments. Language standards C. Lit. [1, pp.7-18; 2, pp. 13-20; 5-14]			№3.		
7	The structure of the language program C. Standard functions The structure of the Pascal language program. Prototypes of functions. Guidelines for including the contents of files. Basic input-output tools. Function output specifications. Data entry using standard functions. Lit. [1, pp.19-30; 2, pp. 27-48; 5-14]	Arrays. Working with one-dimensional arrays.		Elaboration of lecture material. Preparation for laboratory work №4 and practical work №4. Independent work on program development for laboratory work №4.	9	[1, C.19-30; 2, C. 27-48; 5-14]
8	General information about operators General information about operators. Folded operator. Empty operator. Expressions. Operation sign. Types of arithmetic operations. Priority of operations. Lit. [1, pp.31-111; 2, pp. 17-26; 5-14]	Arrays. Working with one-dimensional arrays.	Branched algorithms. Implementation in the C language.	Elaboration of lecture material. Preparation for laboratory work №4 and practical work №4. Independent work on program development for laboratory work №4.	9	[1, C.31-111; 2, C. 17-26; 5-14]
9	General information about operators Branching operator, its forms. Logical	Two-dimensional arrays.	Programming of cyclic algorithms. Implementation in the C language.	Elaboration of lecture material. Preparation for laboratory work №5 and practical work №5. Independent work on	9	[1, C.85-111; 2, C. 17-27; 5-14]

	<p>operations. Compiled assignment operators. Conditional selection operation. Lit. [1, pp.85-111; 2, pp. 17-27; 5-14]</p>			<p>program development for laboratory work №5.</p>		
10	<p>General information about operators Using a comma. Loop operators. Operator with parameter and its features. Operators of cycle-while and cycle-do, their differences. Lit. [1, pp.85-111; 2, pp. 17-27; 4-14]</p>	Two-dimensional arrays.	<p>Arrays. Working with one-dimensional arrays. Implementation in the C language.</p>	<p>Elaboration of lecture material. Preparation for laboratory work №5 and practical work №5. Independent work on program development for laboratory work №5.</p>	9	[1, C.85-111; 2, C. 17-27; 4-14]
11	<p>Functions in the C language Announcement and definition of functions. Arguments, parameters, examples. Transfer parameters by value and by link. Return from function and return values. Arithmetic functions. Recursion. Recursive calls. Direct and indirect recursion. Basic principles of structural programming. Different types of functions. Lit. [1, pp.183-234; 2, pp. 27-47; 4-14]</p>	Functions.		<p>Elaboration of lecture material. Preparation for laboratory work №6 and practical work №6. Independent work on program development for laboratory work №6.</p>	9	[1, C.183-234; 2, C. 27-47; 4-14]
12	<p>Arrays Description of arrays. Arrays of</p>	Functions.	Two-dimensional arrays. Implementation in	<p>Elaboration of lecture material. Preparation for laboratory work</p>	9	[1, C.125-140; 2, C. 29; 3, C.44-59, 105-

	<p>elements. Array declarations. Access to array components, one- and n-dimensional arrays. Examples. Sorting and search methods. Computational complexity of algorithms. Lit. [1, pp.125-140; 2, pp. 29; 3, pp.44-59, 105-117; 4-14]</p>		the C language.	<p>№6 and practical work №6. Independent work on the development of a program for laboratory work №6.</p>		117; 4-14]
13	<p>Strings Strings. Concepts and announcements. Indexes of string elements. Merge strings. Comparison of lines. Input - output lines. Other functions for string processing. Examples. Lit. [1, pp.141-163; 2, pp. 52-53; 4-14]</p>	Strings	<p>Functions. Implementation in the C language</p>	<p>Elaboration of lecture material. Preparation for laboratory work 7 and practical work №7. Independent work on the development of a program for laboratory work №7.</p>	9	[1, C.311-362; 2, C. 54-60; 4-14]
14	<p>Files Files. Definition. File types. General rules for all file types. Text files. Algorithm for creating a text file, reading, writing, writing to a text file. Features of application of standard functions and procedures when working with files. Lit. [1, pp.311-362; 2, pp. 54-60; 4-14]</p>	Strings	<p>Strings. Implementation in the C language</p>	<p>Elaboration of lecture material. Preparation for laboratory work №7 and practical work №7. Independent work on the development of a program for laboratory work №7.</p>	9	[1, C.311-362; 2, C. 54-60; 4-14]
15	<p>Pointers. The concept of pointers.</p>	Streams and files. Working with text files. Structures.		<p>Elaboration of lecture material. Preparation for laboratory work</p>	9	[1, C.112-124; 2, C. 34-35; 4-14]

	<p>Creating pointers. Announcement of pointers. Pointers and types of variables. Pointers and arrays. Transfer arrays to functions. Dynamic data structures. Lit. [1, pp.112-124; 2, pp. 34-35; 4-14]</p>			<p>№8 and practical work №8. Independent work on the development of a program for laboratory work №8.</p>		
16	<p>Structures, associations and non-standard data types. The simplest structures. Complex structures. Arrays of structures. Lit. [1, pp.112-124; 2, pp. 30-31; 4-14]</p>	<p>Streams and files. Working with text files. Structures.</p>	<p>Streams and files. Working with text files. Their implementation in the C language.</p>	<p>Elaboration of lecture material. Preparation for laboratory work №8 and practical work №8. Independent work on the development of a program for laboratory work №8.</p>	9	<p>[1, C.112-124; 2, C. 30-31; 4-14]</p>
17	<p>Additional information about pointers Pointers to pointers. The difference between dynamic and static data. Stack, queue, binary tree. Lit. [1, pp.246-295; 2, pp. 35; 4-14]</p>	Final class	Final class	Final class	11	<p>[1, C.246-295; 2, C. 35; 4-14]</p>

Note: * Lectures, practical classes last two hours; the sequence of classes is determined by the schedule (may not correspond to the numbered weeks)

Course policy.

The organization of the educational process in the course meets the requirements of the provisions on the organizational and educational support of the educational process, educational program and curriculum. The student is obliged to attend lectures, practical and laboratory classes according to the schedule, not to be late for classes, to perform tasks according to the schedule. The student is obliged to complete the missed practical or laboratory lesson in full and report to the teacher no later than one week before the next certification. The student must prepare for practical classes on the relevant topic and be active. Acquisition by a person of knowledge of the discipline or its separate sections in non-formal education is credited in accordance with the Regulations on the procedure for re-enrollment of learning outcomes in KhNU (<http://khnu.km.ua/root/files/01/06/03/006.pdf>).

Learning outcomes assessment criteria.

Current control is carried out during lectures, practical and laboratory classes, as well as on the days of control activities established by the work plan of the discipline. Semester tests are conducted in the form of exam, credit and defense of the course project. At the same time, the results of the current control are taken into account when deriving the final assessment.

The teaching of the discipline uses such types of classes as lectures, laboratory work, practical work, individual counseling and guidance of independent student work.

Each type of work required for the discipline is graded on a four-point scale. The semester final grade is defined as the weighted average of all types of academic work performed and passed positively, taking into account the weighting factor. Weights vary depending on the structure of the discipline and the importance of its individual types of work. A student who scored a positive weighted average score for the current work and did not pass the exam is considered to have failed.

When assessing students' knowledge, various means of control are used, in particular: oral examination before admission to laboratory work - is carried out at the beginning; mastering the theoretical material on the topic is checked by test control; the quality of performance, acquisition of theoretical knowledge and practical skills is checked by defending each laboratory work in accordance with the work program of the discipline and the working curriculum.

The assessment, which is set for the laboratory lesson, consists of the following elements: oral examination of students before admission to laboratory work; knowledge of theoretical material on the topic; the quality of the protocol and graphic part; the student's ability to justify constructive decisions; timely protection of laboratory work. To complete the discipline program, the student must receive 12 (autumn semester) and 8 (spring semester) grades for laboratory work in each semester.

The term of defense of laboratory work is considered timely if the student defended it in the next lesson after the work.

The student must complete the missed laboratory lesson in the laboratories of the department within the period set by the teacher with registration in the relevant journal of the department, but not later than two weeks before the end of theoretical classes in the semester.

When assessing students' knowledge, the teacher is guided by the following criteria.

The student receives an "excellent" grade on the ECTS-A scale (see grading scale) for deep and complete mastery of the content of educational material in which he is easily oriented, the conceptual apparatus, the ability to connect theory with practice, solve practical problems, express and substantiate their judgments. Excellent assessment involves a competent, logical presentation of the answer (both orally and in writing), quality exterior design. The student must acquire practical skills in compiling various algorithms and developing programs based on these algorithms. The grade "excellent" is given to a student who has deeply mastered the operators, functions and procedures of the C language and is able to rationally apply them, knows the techniques and is able to use them in compiling algorithms and programs. The student should not hesitate to change the question, should make detailed and generalizing conclusions.

The student receives a grade of "good" on the ECTS-B scale for full mastering of the study material, mastery of the conceptual apparatus, orientation in the studied material, conscious use of knowledge to solve practical problems, competent presentation of the answer, but in the content and form of the answer (errors), vague wording of patterns, etc. The student's answer should be based on independent thinking.

The student receives a grade of "good" on the ECTS - C scale for the correct answer with one significant error.

Grades "satisfactory", according to the ECTS - D scale, deserves a student who has shown knowledge of the basic curriculum in the amount necessary for further study and practical activities in the profession that copes with the practical tasks provided by the program. As a rule, the student's answer is based on the level of reproductive thinking, the student knows little about the structure of the course, makes mistakes in the answer, mastered and acquired practical skills in compiling programs, but made inaccuracies. He hesitates in answering the modified question, however, the student has the knowledge that allows him under the guidance of the teacher to eliminate inaccuracies in the answer.

Grades "satisfactory", according to the ECTS - E scale, the student deserves for incomplete mastery of the program material, but he gained knowledge and acquired practical skills in developing programs in the C language.

The grade "unsatisfactory", according to the ECTS - FX scale, is given when the student has disparate, unsystematic knowledge, can not distinguish between primary and secondary, errors in defining concepts, distorts their meaning, chaotically and uncertainly presents material, can not use knowledge in solving practical tasks. As a rule, a grade of "unsatisfactory" is given to a student who cannot continue his studies without additional knowledge of the course.

The grade "unsatisfactory", according to the ECTS - F scale, is given to the student for complete ignorance and misunderstanding of the study material or refusal to answer and involves re-learning the student in the discipline.

Each type of work is evaluated on a four-point scale. The semester final grade is defined as the weighted average of all types of work.

**Structuring the course by types of work and assessing learning outcomes
for full-time students in the semester according to weighing coefficients**

Classwork												Independent, individual work						Semester control (Pass/ fail test)			
I semester																					
Laboratory work №:				Practical classes №									Test control:	Tests		Exam					
1,2	3,4	5,6	7,8	9	10	11	12	1	2	3	4	5	6	7	8	9	T 1-17	T 1-8	T 9 - 12	T 13 - 17	
WC: 0,2				WC: 0									0,1	0,1	0,1	0,1	0,4				

Abbreviations: T – topic of the course; WC – weighing coefficient.

Assessing tests. The thematic test for each student consists of twenty-five test items, each of which is graded one point. The maximum amount of points that a student can score is 25.

The assessment is based on a four-point scale.

The correspondence of the scored points for the test to the grade given to the student is presented in the table below.

Scored points for the test	1–11	12–14	15–22	23-25
Grade	2	3	4	5

30 minutes are allotted for testing. Testing is performed using a modular learning environment MOODLE. The student registers the correct answers online in the modular environment MOODLE. After 30 minutes, students complete the test and send their answers to the server. The teacher announces the results of testing according to the journal of assessments of the modular environment MOODLE.

If a student receives a negative grade, he must retake it in the prescribed manner, but always before the next control.

The final semester grade according to the national scale and the ECTS scale is set in an automated mode after entering all the grades in the electronic journal. The ratios of the domestic assessment scale and the ECTS assessment scale are given in the following table.

To move from the national grade to the ECTS scale, you need to find the arithmetic mean of the national scale, multiply it by the appropriate weighting factor and, adding all the components, get the sum of points that will determine a particular ECTS grade.

Correspondence of the national and ECTS grading scales

<i>ECTS grade</i>	<i>Institutional score scale</i>	<i>Institutional grade</i>	<i>Assessment criteria</i>	
A	4,75-5,00	5	Passed	Excellent – deep and complete mastery of educational material and demonstrating relevant skills and abilities.
B	4,25-4,74	4		Good – complete knowledge of the material with a few minor errors.
C	3,75-4,24	4		Good – correct answer in general with two to three significant errors.
D	3,25-3,74	3		Satisfactory – incomplete mastery of the program material but sufficient for practical activities in the professional field.
E	3.00-3,24	3		Satisfactory – incomplete mastery of the program material that meets the minimum assessment criteria.
FX	2,00-2,99	2	Failed	Unsatisfactory – unsystematic knowledge and inability to continue studies without additional knowledge of the course.
F	0,00-1,99	2		Unsatisfactory – serious further work is needed and the course is to be retaken.

Questions for the final test in the course

1. Information technologies and systems
2. History of computer technology
3. History of operating systems, their main common implementations
4. Information and its presentation
5. Binary arithmetic
6. General structure of the computer
7. Computer systems and their components
8. The concept of algorithms and their properties
9. Step-by-step detailing method
10. Recording of the algorithm in the form of block diagrams
11. Forms of representation of algorithms
12. Types of basic structures of algorithms
13. Alphabet and dictionary of language C
14. Basic data types
15. Constants
16. Identifiers
17. Keywords
18. Comments
19. Language standards C
20. The structure of the language program C
21. #include Directive
22. The main means of input-output
23. General information about operators
24. Expressions
25. Branching operator
26. Logical operations
27. Compound assignment operators
28. Conditional selection operation
29. Using a comma
30. Cycle operators
31. Cycle with postcondition, example program
32. FOR cycle
33. Cycle with a prerequisite, an example of a program
34. Announcement and definition of functions
35. Recursive call
36. Arrays of elements
37. Multidimensional arrays
38. The method of pairwise permutation of elements
39. The method of the smallest elements
40. Binary search method
41. The concept of pointers
42. Creating pointers
43. Announcement of pointers
44. Pointers and types of variables
45. Pointers and arrays
46. Transfer of arrays in functions
47. The concept of strings
48. Ways to allocate memory
49. Functions of input and output of characters and lines
50. Increment and decrement operations
51. Operator continue
52. Break operator
53. Return the values of the function using the return operator
54. Prototypes
55. Additional assignment operations
56. Object code files, executables, and libraries
57. IF branching operator
58. Compound literals.
59. Interaction with files.
60. Standard files.
61. The fopen () function.

- 62. File compression program.
- 63. Input - output files.
- 64. Random access to files
- 65. The simplest data structures
- 66. Complex structures.
- 67. Arrays of structures.
- 68. Initialization of structures.

TEACHING AND LEARNING MATERIALS

The educational process in the course is provided with all necessary educational materials and guidelines in MOODLE environment.

RECOMMENDED LITERATURE

1. Seacord R. Effective C: An Introduction to Professional C Programming. - San Francisco, California, No Starch Press, 2020. - 272 p.
2. Gustedt J. Modern C. - Shelter Island, Manning, 2020. – 408 p.
3. King K.N. C programming: a modern approach. - W. W. Norton & Company, 2020. – 832 p.
4. Harwani B.M. Practical C Programming: Solutions for modern C developers to create efficient and well-structured programs. – Birmingham, Packt Publishing, 2020. – 616 p.
5. Hubert H.W. Intermediate C programming for the PIC microcontroller: simplifying embedded programming. - Berkeley, CA, Apress L. P., 2020. – 318 p.
6. Winkle L.Van. Hands-On Network Programming with C: Learn socket programming in C and write secure and optimized network code. – Birmingham, Packt Publishing, 2019. – 478 p.
7. Harwani B.M. C programming cookbook: over 40 recipes exploring data structures, pointers, interprocess communication, and database in C. - Birmingham, UK, Packt Publishing, 2019. – 344 p.
8. Joyce P. Numerical C: applied computational programming with case studies. - Berkeley, CA, Apress, 2019. – 312 p.
9. McGrath M. C Programming in easy steps: Updated for the GNU Compiler version 6.3.0 and Windows 10 5th Edition. - In Easy Steps Limited, 2018. – 192 p.
10. Kane W, C programming for business. Book one, Introduction to problem solving in C language. - Choice Publishing, 2017. – 284 p.

Program creator:



Dr. Sc., prof. V.V. Martynyuk

Approved:

Head of the department:



Dr. Sc., prof. L.P. Bedratyuk

Guarantor EP «SE»:



PhD, assoc. prof. L.P. Bedratyuk

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет програмування та комп'ютерних і телекомунікаційних систем
Кафедра інженерії програмного забезпечення

Декан ФІТ  ЗАТВЕРДЖУЮ
Говорушченко Т.О.
1 вересня 2024 р.



СИЛАБУС

Навчальна дисципліна Програмування

Освітньо-професійна програма Інженерія програмного забезпечення

Рівень вищої освіти перший (бакалаврський)

Загальна інформація

Позиція	Зміст інформації
Викладач	Мартинюк Валерій Володимирович
Профайл викладача	http://ki.khnu.km.ua/team/martyniuk-valerij-volodymyrovych/
Е-mail викладача	martyniukvv@khnu.km.ua
Контактний телефон	заповнюється за домовленістю
Сторінка дисципліни в ІСУ	https://msn.khmnu.edu.ua/enrol/index.php?id=8709
Навчальний рік	2024-2025
Консультації	Очні: середа, 6-а пара, 4-325; п'ятниця, 6-а пара, 4-325; онлайн: за необхідністю та попередньою домовленістю

Характеристика дисципліни

Форма здобуття освіти	Курс	Семестр	Загальне навантаження		Кількість годин						Форма семестрового контролю			
			Кредити ЄКТС	Години	Аудиторні заняття				Індивідуальна робота студента	Самостійна робота, в т.ч. РС	Курсовий проект	Курсова робота	Залік	Іспит
					Разом	Лекції	Лабораторні роботи	Практичні заняття						
ОД	1	1	10	300	136	34	68	34		164			+	
Разом			10	300	136	34	69	70		164			1	

Анотація дисципліни

В дисципліні планується освоєння студентами матеріалу з основ програмування, визначення для вирішення задач технології програмування, мови, системи програмування, інструментального середовища, здійснення функціональної та об'єктної декомпозиції програми відповідно до обраної технології програмування, виконання розробки коду програми, виправлення синтаксичних та семантичних помилок та рефакторинг коду, налагоджування та тестування програми, програмування динамічних структур даних, оброблення виключень.

Дисципліна викладається для студентів денної форми навчання спеціальностей галузі інформаційних технологій. При викладанні дисципліни використовуються активні і творчі форми проведення занять. Завдання дисципліни передбачає надання студентам знань про алгоритмізацію та її використання для розв'язування практичних задач, навчити програмувати мовою C/C++.

Пререквізити: вища математика; дискретна математика; **кореквізити:** об'єктно-орієнтоване програмування, структури даних і алгоритми.

Мета дисципліни: 1) формування компетентностей, необхідних для абстрактного мислення, аналізу та синтезу при здійсненні алгоритмізації та програмуванні; 2) розвиток у студентів фахового стилю мислення з алгоритмізації; 3) надання знань з теорії алгоритмів та конструкцій мов програмування (на прикладі мови C), необхідних для подальшого вивчення спеціальних дисциплін та для практичної інженерної діяльності; 4) вироблення у студентів вміння використовувати набуті знання при розробці програм.

Завдання дисципліни. Надати студентам знання про алгоритмізацію та її використання для розв'язування практичних задач, навчити програмувати мовою C/C++.

Очікувані результати навчання.

Студент, який успішно завершив вивчення дисципліни, повинен оволодіти знаннями про основи алгоритмізації та програмування, складати алгоритми та програмувати їх з використанням мови C/C++, парадигми програмування, фундаментальні структури даних, структурне програмування, конструкції мов програмування, рекурсію, програмування динамічних структур даних, алгоритми та структури даних; обробку стандартних і нестандартних виключень, моделювати, проектувати та розробляти алгоритми, оцінювати їх складність; планувати та ефективно організовувати роботу при складанні програм мовою C.

Тематичний і календарний план вивчення дисципліни

Осінній семестр

№ тижня	Тема лекцій*	Тема практичного заняття*	Тема лабораторної роботи*	Самостійна робота студентів		
				Зміст	Год.	Література
1	2	3	4	5	6	7
1	Вступ. Інформаційні технології Вступ. Інформація та її подання. Двійкова арифметика. Літ. [15, С.14-40]	Лінійні алгоритми.	Лінійні алгоритми. Реалізація мовою C.	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №1 та практичної роботи №1. Самостійна робота над розробкою програми до лабораторної роботи №1.	9	[15, С.14-40]
2	Вступ. Комп'ютерні системи Загальна структура комп'ютера. Комп'ютерні системи та їх складові. Прикладне програмне забезпечення. Системне програмне забезпечення. Парадигми програмування. Літ. [15, С.14-40]	Лінійні алгоритми.	Лінійні алгоритми. Реалізація мовою C.	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №1 та практичної роботи №1. Самостійна робота над розробкою програми до лабораторної роботи №1.	9	[15, С.14-40]
3	Алгоритми та їх властивості. Основні поняття Алгоритми	Розгалужені алгоритми.	Розгалужені алгоритми. Реалізація мовою C.	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи	9	[3, С.12-27]

	алгоритмізація. Поняття алгоритму. Алгоритми та їх властивості. Поняття про алгоритми та їх властивості. Властивості алгоритму. Метод покрокової деталізації. Форми подання алгоритмів. Літ. [3, С.12-27]			№2 та практичної роботи №2. Самостійна робота над розробкою програми до лабораторної роботи №2.		
4	Алгоритми та їх властивості. Типи алгоритмів Запис алгоритму у вигляді блок-схем. Приклади. Базові алгоритмічні структури. Слідування. Розгалуження. Повторення. Приклади. Типи основних структур алгоритмів. Поняття програми. Літ. [3, С.12-27]	Розгалужені алгоритми.	Розгалужені алгоритми. Реалізація мовою С.	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №2 та практичної роботи №2. Самостійна робота над розробкою програми до лабораторної роботи №2.	9	[3, С.12-27]
5	Введення в С Вступ до мови програмування С. Алфавіт та словник мови. Ідентифікатори. Базові типи даних. Модифікатори типів. Літ. [1, С.7-18; 2, С. 6-12; 5-14]	Програмування циклічних алгоритмів.	Програмування циклічних алгоритмів. Реалізація мовою С.	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №3 та практичної роботи №3. Самостійна робота над розробкою програми до лабораторної роботи №3.	9	[1, С.7-18; 2, С. 6-12; 5-14]
6	Структура програми мови С Константи. Префіксна та суфіксна форми. Вирази та операції. Десятькове, вісімкове та шістнадцятькове представлення. Ідентифікатори. Ключові слова. Коментарі. Стандарти мови С. Літ. [1, С.7-18; 2, С. 13-20; 5-14]	Програмування циклічних алгоритмів.	Програмування циклічних алгоритмів. Реалізація мовою С.	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №3 та практичної роботи №3. Самостійна робота над розробкою програми до лабораторної роботи №3.	9	[1, С.7-18; 2, С. 13-20; 5-14]
7	Структура програми мови С. Стандартні функції Структура програми мови Паскаль. Прототиби функцій.	Масиви. Робота з одним ірними масивами.	Масиви. Робота з одновимірними масивами. Реалізація мовою С.	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №4 та практичної роботи №4. Самостійна	9	[1, С.19-30; 2, С. 27-48; 5-14]

	Директиви для включення вмісту файлів. Основні засоби введення-виведення. Специфікації виведення функцій. Введення даних з використанням стандартних функцій. Літ. [1, С.19-30; 2, С. 27-48; 5-14]			робота над розробкою програми до лабораторної роботи №4.		
8	Загальні відомості про оператори Загальні відомості про оператори. Складений оператор. Порожній оператор. Вирази. Знак операції. Типи арифметичних операцій. Пріоритет операцій. Літ. [1, С.31-111; 2, С. 17-26; 5-14]	Масиви. Робота з одним ірними масивами.	Масиви. Робота з одномірними масивами. Реалізація мовою С.	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №4 та практичної роботи №4. Самостійна робота над розробкою програми до лабораторної роботи №4.	9	[1, С.31-111; 2, С. 17-26; 5-14]
9	Загальні відомості про оператори Оператор розгалуження, його форми. Логічні операції. Складені оператори присвоювання. Операція вибору за умовою. Літ. [1, С.85-111; 2, С. 17-27; 5-14]	Двовимірні масиви.	Двовірні масиви. Реалізація мовою С.	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №5 та практичної роботи №5. Самостійна робота над розробкою програми до лабораторної роботи №5.	9	[1, С.85-111; 2, С. 17-27; 5-14]
10	Загальні відомості про оператори Використання коми. Оператори циклу. Оператор з параметром та його особливості. Оператори циклу-поки та циклу-до, їх відмінності. Літ. [1, С.85-111; 2, С. 17-27; 4-14]	Двовимірні масиви.	Двовірні масиви. Реалізація мовою С.	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №5 та практичної роботи №5. Самостійна робота над розробкою програми до лабораторної роботи №5.	9	[1, С.85-111; 2, С. 17-27; 4-14]
11	Функції в мові С Оголошення і визначення функцій. Аргументи, параметри, приклади. Передача параметрів за значенням та за посиланням.	Функції.	Функції. Реалізація мовою С.	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №6 та практичної роботи №6. Самостійна робота над розробкою програми до	9	[1, С.183-234; 2, С. 27-47; 4-14]

	Повернення з функції та повернення значень. Арифметичні функції. Рекурсія. Рекурсивні виклики. Пряма та непряма рекурсії. Основні принципи структурного програмування. Різні типи функцій. Літ. [1, С.183-234; 2, С. 27-47; 4-14]			лабораторної роботи №6.		
12	Масиви Опис масивів. Масиви елементів. Оголошення масивів. Доступ до компонентів масиву, одно- і n-мірні масиви. Приклади. Методи сортування та пошуку. Обчислювальна складність алгоритмів. Літ. [1, С.125-140; 2, С. 29; 3, С.44-59, 105-117; 4-14]	Функції.	Функції. Реалізація мовою С.	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №6 та практичної роботи №6. Самостійна робота над розробкою програми до лабораторної роботи №6.	9	[1, С.125-140; 2, С. 29; 3, С.44-59, 105-117; 4-14]
13	Рядки Рядки. Поняття та оголошення. Індеси елементів рядка. Об'єднання рядків. Порівняння рядків. Введення – виведення рядків. Інші функції для обробки рядків. Приклади. Літ. [1, С.141-163; 2, С. 52-53; 4-14]	Рядки.	Рядки. Реалізація мовою С.	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи 7 та практичної роботи №7. Самостійна робота над розробкою програми до лабораторної роботи №7.	9	[1, С.141-163; 2, С. 52-53; 4-14]
14	Файли Файли. Визначення. Типи файлів. Загальні правила для всіх типів файлів. Текстові файли. Алгоритм створення текстового файлу, читання, запис, дозапис в текстовий файл. Особливості застосування стандартних функцій і процедур при роботі з файлами. Літ. [1, С.311-362; 2, С. 54-60; 4-14]	Рядки.	Рядки. Реалізація мовою С.	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №7 та практичної роботи №7. Самостійна робота над розробкою програми до лабораторної роботи №7.	9	[1, С.311-362; 2, С. 54-60; 4-14]

15	Вказівники. Поняття вказівників. Створення вказівників. Оголошення вказівників. Вказівники і типи змінних. Вказівники і масиви. Передача масивів в функції. Динамічні структури даних. Літ. [1, С.112-124; 2, С. 34-35; 4-14]	Потоки та файли. Робота з текстовими файлами. Структури.	Потоки та файли. Робота з текстовими файлами. Їх реалізація мовою С.	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №8 та практичної роботи №8. Самостійна робота над розробкою програми до лабораторної роботи №8.	9	[1, С.112-124; 2, С. 34-35; 4-14]
16	Структури, об'єднання і нестандартні типи даних. Найпростіші структури. Складні структури. Масиви структур. Літ. [1, С.112-124; 2, С. 30-31; 4-14]	Потоки та файли. Робота з текстовими файлами. Структури.	Потоки та файли. Робота з текстовими файлами. Їх реалізація мовою С.	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №8 та практичної роботи №8. Самостійна робота над розробкою програми до лабораторної роботи №8.	9	[1, С.112-124; 2, С. 30-31; 4-14]
17	Додаткові відомості про вказівники Вказівники на вказівники. Відмінність динамічних даних від статичних. Стек, черга, двійкове дерево. Літ. [1, С.246-295; 2, С. 35; 4-14]	Підсумкове заняття.	Підсумкове заняття.	Підсумкове заняття.	11	[1, С.246-295; 2, С. 35; 4-14]

Примітка: * Лекції, практичні і лабораторні заняття проводяться по дві години; послідовність проведення занять визначається розкладом (може не відповідати нумерованим тижням)

Політика дисципліни.

Організація освітнього процесу з дисципліни відповідає вимогам положень про організаційне і навчально-методичне забезпечення освітнього процесу, освітній програмі та навчальному плану. Студент зобов'язаний відвідувати лекції, практичні та лабораторні заняття згідно з розкладом, не запізнюватися на заняття, завдання виконувати відповідно до графіка. Пропущене практичне чи лабораторне заняття студент зобов'язаний опрацювати самостійно у повному обсязі і відзвітувати перед викладачем не пізніше, ніж за тиждень до чергової атестації. До практичних занять студент має підготуватися за відповідною темою і проявляти активність. Набуті особою знання з дисципліни або її окремих розділів у неформальній освіті зараховуються відповідно до Положення про порядок перезарахування результатів навчання у ХНУ (<http://khnu.km.ua/root/files/01/06/03/006.pdf>).

Критерії оцінювання результатів навчання.

Поточний контроль здійснюється під час лекційних, практичних та лабораторних занять, а також у дні проведення контрольних заходів, встановлених робочим планом дисципліни. Семестровий контроль проводиться у формі іспиту. При цьому при виведенні остаточної оцінки враховуються результати поточного контролю.

При викладанні дисципліни використовуються такі види навчальних занять, як лекції, лабораторні роботи,

практичні роботи, індивідуальне консультування і керівництво самостійною роботою студента.

Кожний, обов'язковий для оцінювання, вид роботи з дисципліни оцінюється за *чотирибальною* шкалою. Семестрова підсумкова оцінка визначається як середньозважена з усіх видів навчальної роботи, виконаних і зданих *позитивно* з врахуванням коефіцієнта вагомості. Вагові коефіцієнти змінюються залежно від структури дисципліни і важливості окремих її видів робіт. Студент, який набрав позитивний середньозважений бал за поточну роботу і не здав підсумковий контрольний захід (іспит), вважається нествигаючим.

При оцінюванні знань студентів використовуються різні засоби контролю, зокрема: усне опитування перед допуском до виконання лабораторної роботи – здійснюється на її початку; засвоєння теоретичного матеріалу з тем перевіряється тестовим контролем; якість виконання, набуття теоретичних знань і практичних навичок перевіряється шляхом захисту кожної лабораторної роботи згідно з робочою програмою дисципліни і робочим навчальним планом.

Оцінка, яка виставляється за *лабораторне заняття*, складається з таких елементів: усне опитування студентів перед допуском до виконання лабораторної роботи; знання теоретичного матеріалу з теми; якість оформлення протоколу і графічної частини; вміння студента обґрунтувати прийняті конструктивні рішення; своєчасний захист лабораторної роботи. Для виконання програми дисципліни студент повинен отримати 12 (осінній семестр) і 8 (весняний семестр) оцінок за лабораторні роботи в кожному з семестрів.

Термін захисту лабораторної роботи вважається своєчасним, якщо студент захистив її на наступному після виконання роботи занятті. За несвоєчасний захист лабораторної роботи з неважкої причини студент за позитивну відповідь отримує оцінку «задовільно».

Пропущене лабораторне заняття студент повинен відпрацювати в лабораторіях кафедри у встановлений викладачем термін з реєстрацією у відповідному журналі кафедри, але не пізніше, ніж за два тижні до кінця теоретичних занять у семестрі.

При оцінюванні знань студентів викладач керується такими критеріями.

Оцінку „відмінно”, за шкалою ECTS – A (див. шкалу оцінок), отримує студент за глибоке і повне опанування змісту навчального матеріалу, в якому він легко орієнтується, понятійного апарату, за уміння зв'язувати теорію з практикою, вирішувати практичні завдання, висловлювати і обґрунтовувати свої судження. Відмінна оцінка передбачає грамотний, логічний виклад відповіді (як в усній, так і в письмовій формі), якісне зовнішнє оформлення. Студент повинен набутти практичних навичок із складання різних алгоритмів та розробки програм за цими алгоритмами. Оцінка "відмінно" виставляється студенту, який глибоко засвоїв оператори, функції та процедури мови C та вмів їх раціонально застосувати, знає методики та вмів ними користуватися при складанні алгоритмів та програм. Студент не повинен вагатися при видозміні запитання, повинен робити детальні та узагальнюючі висновки.

Оцінку „добре”, за шкалою ECTS – B, отримує студент за повне засвоєння навчального матеріалу, володіння понятійним апаратом, орієнтування в вивченому матеріалі, свідоме використання знань для вирішення практичних завдань, грамотний виклад відповіді, але у змісті і формі відповіді мали місце окремі неточності (похибки), нечіткі формулювання закономірностей тощо. Відповідь студента повинна будуватись на основі самостійного мислення.

Оцінку „добре”, за шкалою ECTS – C, отримує студент за правильну відповідь з однією суттєвою помилкою.

Оцінки "задовільно", за шкалою ECTS – D, заслуговує студент, який виявив знання основного навчально-програмного матеріалу в обсязі, необхідному для подальшого навчання та практичної діяльності за професією, що справляється з виконанням практичних завдань, передбачених програмою. Як правило, відповідь студента будується на рівні репродуктивного мислення, студент слабо знає структуру курсу, допускає помилки у відповіді, засвоїв і набув практичних навичок у складанні програм, але допустив неточності. Вагається при відповіді на видозмінене запитання, разом з тим студент володіє знаннями, що дозволяють йому під керівництвом викладача усунути неточності у відповіді.

Оцінки "задовільно", за шкалою ECTS – E, заслуговує студент за неповне опанування програмного матеріалу, але ним отримані знання і набуті практичні навички із розробки програм мовою C.

Оцінка „незадовільно”, за шкалою ECTS – FX, виставляється, коли студент має розрізнені, безсистемні знання, не вмів виділяти головне і другорядне, допускається помилок у визначенні понять, перекичує їх зміст, хаотично і невпевнено викладає матеріал, не може використовувати знання при вирішенні практичних завдань. Як правило, оцінка "незадовільно" виставляється студенту, який не може продовжити навчання без додаткових знань з курсу.

Оцінка „незадовільно”, за шкалою ECTS – F, виставляється студенту за повне незнання і нерозуміння навчального матеріалу або відмову від відповіді і передбачає повторне навчання студента з дисципліни.

Кожний вид роботи оцінюється за чотирибальною шкалою. Семестрова підсумкова оцінка визначається як середньозважена з усіх видів робіт.

Структурування дисципліни за видами робіт і оцінювання результатів навчання студентів у семестрі за ваговими коефіцієнтами

Аудиторна робота												Самостійна, індивідуальна робота									Форма семестрового контролю		
I семестр																							
Лабораторні роботи №:				Практичні роботи №								Тестовий контроль:		КР		Іспит							
1,2	3,4	5,6	7,8	9	10	11	12	1	2	3	4	5	6	7	8	9	Т 1-17		Т 1-8		Т 9 - 17		

ВК:	0,3	ВК:	0	0,1	0,1	0,1	0,4
-----	-----	-----	---	-----	-----	-----	-----

Примітка: Т – тема дисципліни; ВК – ваговий коефіцієнт;

Оцінювання тестових завдань. Тематичний тест для кожного студента складається з двадцяти п'яти тестових завдань, кожне з яких оцінюється одним балом. Максимальна сума балів, яку може набрати студент, складає 25.

Оцінювання здійснюється за чотирибальною шкалою.

Відповідність набраних балів за тестове завдання оцінці, що виставляється студенту, представлена у нижченаведеній таблиці.

Сума балів за тестове завдання	1–11	12–14	15–22	23–25
Оцінка	2	3	4	5

На тестування відводиться 30 хвилин. Тестування проводиться з використанням модульного середовища для навчання MOODLE. Правильні відповіді студент реєструє в он-лайн режимі в модульному середовищі MOODLE. Через 30 хвилин студенти завершують тестування та надсилають свої відповіді на сервер. Викладач оголошує результати тестування згідно журналу оцінок модульного середовища MOODLE.

Якщо студент отримав негативну оцінку, то він має перездати її в установленому порядку, але обов'язково до терміну наступного контролю. У випадку, коли студент не виконав індивідуальний план з дисципліни у заплановані терміни без поважних причин, то під час відпрацювання заборгованості при позитивній відповіді йому виставляється оцінка „задовільно”.

Підсумкова семестрова оцінка за національною шкалою і шкалою ЄКТС встановлюється в автоматизованому режимі після внесення усіх оцінок до електронного журналу. Співвідношення вітчизняної шкали оцінювання і шкали оцінювання ЄКТС наведені у наступній таблиці.

Для переходу від вітчизняної оцінки до оцінки за шкалою ECTS необхідно знайти середньоарифметичну оцінку за вітчизняною шкалою, помножити її на відповідний ваговий коефіцієнт і, додавши всі складові, отримаємо суму балів, які визначають конкретну оцінку ECTS.

Співвідношення вітчизняної шкали оцінювання і шкали оцінювання ЄКТС

Оцінка ECTS	Бали	Вітчизняна оцінка	
A	4,75-5,00	5	ВІДМІННО – глибоке і повне опанування навчального матеріалу і виявлення відповідних умінь та навиків
B	4,25-4,74	4	ДОБРЕ – повне знання навчального матеріалу з кількома незначними помилками
C	3,75-4,24	4	ДОБРЕ – в загальному правильна відповідь з однією суттєвою помилкою
D	3,25-3,74	3	ЗАДОВІЛЬНО – неповне опанування програмного матеріалу, але достатнє для практичної діяльності за професією
E	3,00-3,24	3	ЗАДОВІЛЬНО – неповне опанування програмного матеріалу, що задовольняє мінімальні критерії оцінювання
FX	2,00 -2,99	2	НЕЗАДОВІЛЬНО – безсистемність одержаних знань і неможливість продовжити навчання без додаткових знань з дисципліни
F	0,00-1, 99	2	НЕЗАДОВІЛЬНО – необхідна серйозна подальша робота і повторне вивчення дисципліни

Залік виставляється при отриманні студентом з дисципліни від 3,00 до 5,00 балів. При цьому за вітчизняною шкалою ставиться «зараховано», а за шкалою ECTS – оцінка, що відповідає набраній студентом кількості балів.

Питання для підсумкового контролю з дисципліни

1. Інформаційні технології та системи
2. Історія розвитку комп'ютерної техніки
3. Історія розвитку операційних систем, їх основні поширені реалізації
4. Інформація та її подання
5. Двійкова арифметика
6. Загальна структура комп'ютера
7. Комп'ютерні системи та їх складові
8. Поняття про алгоритми та їх властивості
9. Метод покрокової деталізації
10. Запис алгоритму у вигляді блок-схем
11. Форми подання алгоритмів

12. Типи основних структур алгоритмів
13. Алфавіт та словник мови C
14. Базові типи даних
15. Константи
16. Ідентифікатори
17. Ключові слова
18. Коментарі
19. Стандарти мови C
20. Структура програми мови C
21. Директива #include
22. Основні засоби введення-виведення
23. Загальні відомості про оператори
24. Вирази
25. Оператор розгалуження
26. Логічні операції
27. Складені оператори присвоювання
28. Операція вибору за умовою
29. Використання коми
30. Оператори циклу
31. Цикл з післяумовою, приклад програми
32. Цикл FOR
33. Цикл з передумовою, приклад програми
34. Оголошення і визначення функцій
35. Рекурсивний виклик
36. Масиви елементів
37. Багатомірні масиви
38. Метод попарної перестановки елементів
39. Метод найменших елементів
40. Метод бінарного пошуку
41. Поняття вказівників
42. Створення вказівників
43. Оголошення вказівників
44. Вказівники і типи змінних
45. Вказівники і масиви
46. Передача масивів в функції
47. Поняття про рядки
48. Способи виділення пам'яті
49. Функції введення і виведення символів та рядків
50. Операції інкременту та декременту
51. Оператор continue
52. Оператор break
53. Повернення значень функції за допомогою оператора return
54. Прототипи
55. Додаткові операції присвоювання
56. Файли об'єктного коду, виконувані файли і бібліотеки
57. Оператор розгалуження IF
58. Складені літерали.
59. Взаємодія з файлами.
60. Стандартні файли.
61. Функція fopen().
62. Програма зтискування файлів.
63. Введення – виведення файлів.
64. Довільний доступ до файлів
65. Найпростіші структури даних
66. Складні структури.
67. Масиви структур.
68. Ініціалізація структур.

Основна література

1. Seacord R. Effective C: An Introduction to Professional C Programming. - San Francisco, California, No Starch Press, 2020. - 272 p.
2. Gustedt J. Modern C. - Shelter Island, Manning, 2020. - 408 p.
3. King K.N. C programming: a modern approach. - W. W. Norton & Company, 2020. - 832 p.
4. Harwani B.M. Practical C Programming: Solutions for modern C developers to create efficient and well-structured programs. - Birmingham, Packt Publishing, 2020. - 616 p.
5. Hubert H.W. Intermediate C programming for the PIC microcontroller: simplifying embedded programming. - Berkeley, CA, Apress L. P., 2020. - 318 p.
6. Winkle L.Van. Hands-On Network Programming with C: Learn socket programming in C and write secure and optimized network code. - Birmingham, Packt Publishing, 2019. - 478 p.
7. Harwani B.M. C programming cookbook: over 40 recipes exploring data structures, pointers, interprocess communication, and database in C. - Birmingham, UK, Packt Publishing, 2019. - 344 p.
8. Joyce P. Numerical C: applied computational programming with case studies. - Berkeley, CA, Apress, 2019. - 312 p.
9. McGrath M. C Programming in easy steps: Updated for the GNU Compiler version 6.3.0 and Windows 10 5th Edition. - In Easy Steps Limited, 2018. - 192 p.
10. Kane W, C programming for business. Book one, Introduction to problem solving in C language. - Choice Publishing, 2017. - 284 p.

Додаткова література

1. Harwani B.M. C programming cookbook: over 40 recipes exploring data structures, pointers, interprocess communication, and database in C. - Birmingham, UK, Packt Publishing, 2019. - 344 p.
2. Joyce P. Numerical C: applied computational programming with case studies. - Berkeley, CA, Apress, 2019. - 312 p.
3. McGrath M. C Programming in easy steps: Updated for the GNU Compiler version 6.3.0 and Windows 10 5th Edition. - In Easy Steps Limited, 2018. - 192 p.
4. Kane W, C programming for business. Book one, Introduction to problem solving in C language. - Choice Publishing, 2017. - 284 p.
5. Стандарт C11 <http://www.open-std.org/jtc1/sc22/wg14/www/docs/n1570.pdf>

Інформаційні ресурси

Електронний університет:

1. Модульне середовище для навчання (розміщені усі необхідні матеріали з дисципліни, в тому числі тестові завдання для поточного та семестрового контролю знань).
2. Електронна бібліотека університету

Розробник:



д.т.н., проф. Мартинюк В.В.

Погоджено:

Зав. каф. ІПЗ:



д.ф.-м.н., проф. Бедратюк Л.П.

Гарант ОПП «ІПЗ»:



д.ф.-м.н., проф. Бедратюк Л.П.