ISO 9001 CERTIFIED
SLC. MS.040. LSO9001. 1693

INTERNATIONAL EUROPEAN UNIVERSITY

SYLLABUS

INTERNATIONAL EUROPEAN UNIVERSITY

IT SCHOOL

Fundamentals of programming

2023

SYLLABUS

Course Name	
	Fundamentals of programming
Lecturer (s)	
	Zoia Sherman
Lecturer's profile	
	ttps://it.ieu.edu.ua/pro-yeash/struktura-yeash/kafedra-informatsiinykh-tekhnolohii-v-heodezii-ta-zemleustroi/sklad#zzz-006
Consultations	
online consulting	Monday 3 p.m. – 5 p.m.
offline consulting	Monday 3 p.m. – 5 p.m.
Contact number	
	+380 501811330
E-mail	
	zoiasherman@ieu.edu.ua
Discipline page	

Form of final control	test	def. test	exam
	☐	☐	☒

SYLLABUS

1 Brief discipline annotation

This academic discipline is basic in training students and is aimed at exploring theoretical and methodological fundamentals of designing software in high-level programming languages, taking into account contemporary concepts and trends, mastering tools for creating this software, acquiring practical skills of software development in solving applied problems of different complexity.

2 Background for studying discipline

To successfully learn the Fundamentals of programming discipline, one should have basic knowledge of secondary school Informatics, as well as basic knowledge of such disciplines as Discrete mathematics, Higher mathematics, Analytical geometry and linear algebra.

3 Goal and objectives of the discipline

The **goal** of the Fundamentals of programming discipline is to train highly qualified, competitive specialists in Software engineering proficient in the main procedural and modular programming techniques given contemporary concepts and trends in programming technology. This discipline is basic for a range of programming-related disciplines, such as Object-oriented programming, System programming, Database systems, etc.

Objectives of the discipline:
- to learn theoretical principles and practical techniques in structural, procedural and modular programming;
- to apply key structural constructions of algorithmic programming languages;
- to explore technologies of developing algorithms for applied problems, coding in the chosen programming language;
- to master software debugging and assessment of the validity of the obtained results.

4 Learning outcomes

After learning the discipline, students should
know:
- key structural constructions of algorithmic programming languages;
- technologies of developing algorithms for applied problems;
- approaches to coding in the chosen C/C++ programming language;
- software debugging theory;
- how to assess the validity of the obtained results.

be able to:
- implement linear, branching and cyclic algorithms in C/C++ language;
- implement basic operations for processing one- and two-dimensional static and dynamic arrays in C/C++ language;
- apply arrays as parameters in C/C++ language;
- implement basic operations for processing character arrays in C/C++ language;
- implement algorithms using data structures in C/C++ language;
- implement algorithms using text and binary files in C/C++ language;
- debug software.

5 ECTS credits

4 credits / 120 academic hours

INTERNATIONAL EUROPEAN UNIVERSITY

6 Discipline structure

Content modules and topics	Amount of hours				
	total	including			
		1	p. c.	lab.	i. w.
1	2	3	4	5	6
Content module 1. Basic concepts of programming.					
Topic 1. Introduction. Basic concepts and definitions	4	1	2	-	1
Topic 2. C++ syntax and semantics.	4	1	2	-	1
Topic 3. C++ software structure. Declaring variables	4	1	2	-	1
Topic 4. Controlling input-output streams. Expressions	4	1	2	-	1
Topic 5. C++ language tools for implementing basic algorithm structures. The if operator.	5	2	2	-	1
Topic 6. Loop operators. While Loop.	5	2	2	-	1
Total per content module	26	8	12	-	6
Content module 2. Means of implementing basic structures in programming.					
Topic 7. Do-While Loops.	12	2	4	-	1
Topic 8. For Loops.	12	2	4	-	1
Topic 9. Complex data types in C++. One-dimensional arrays.	12	2	4	-	1
Topic 10. Processing one-dimensional arrays.	12	2	4	-	1
Topic 11. Complex data types in C++. Two-dimensional arrays.	12	2	4	-	1
Topic 12. Processing two-dimensional arrays.	12	2	4	-	1
Total per content module	42	12	24	-	6

6 Discipline structure

Content module 3. Working with I/O functions and term variables.					
Topic 13. Functions. Pointers and references	8	2	4	-	2
Topic 14. Line processing	10	2	6	-	2
Topic 15. Working with std: vector, list containers.	8	2	4	-	2
Topic 16. Working with std: degue, stack, gueue containers.	10	2	6	-	2
Topic 17. Working with std:map, multimap containers.	8	2	4	-	2
Topic 18. File input/output. Working with eof.	8	2	4	-	2
Total per content module	52	12	28	-	12
Total hours	120	32	64		24

7 List of obligatory tasks

1. To learn basic operations in numeral systems.
2. To learn linear, branching and cyclic algorithms in C/C++ language.
3. To implement basic operations for processing one- and two-dimensional static and dynamic arrays in C/C++ language.
4. To apply arrays as parameters in C/C++ language.
5. To implement basic operations for processing character arrays in C/C++ language.
6. To implement algorithms using data structures in C/C++ language.
7. To implement algorithms using text and binary files in C/C++ language.
8. To debug software.

8 List of selective tasks

1. To conduct research of the chosen subject area based on data analysis. Describe the particularities of the examined subject area, core necessary data, the decision making process using the chosen data, required actions to seek hidden knowledge of the subject area in the data according to research objectives.
2. To solve problems on the transformation of logical expressions.
3. To overview free/open source software features.

9 Discipline features

Period of teaching	Semester	International discipline integration	Year of study	Courses: general training/ professional training/elective
2 semesters	1st, 2nd	available	1st year	Professional training course

10 Hardware and software

Personal computer, Windows and Linux OS, MS Visual Studio2019 software development platform.

11 Assessment system and requirements

As part of discipline teaching, one carries out the current and final control of students' knowledge. The final grade is given according to the total rating of students.
The results of the current control of students' knowledge is assessed in general between 0 and 60 scores.
Students are admitted to the final control if they fulfil the requirements of the training program and obtain at least 36 scores for the current learning activity.
Final assessment of students' knowledge is conducted in the form of exam.
The maximum amount of scores that can be obtained during the exam is 40 scores.
The overall score of the discipline is 100. The total grade for the discipline is given according to the national and European scale.

12 Discipline policy

Teaching of the discipline is based on cutting-edge educational technologies aimed at increasing the level of students' interest in the course, providing theoretical and practical knowledge of the discipline.
To activate the learning and cognitive activity of students, the discipline includes the consolidation of knowledge obtained at the lecture and acquisition of practical skills in lecture topics during laboratory classes.

13 Absence policy

Points are not given for missed lectures. If students miss a laboratory work, they should perform all tasks of the missed laboratory work before the next laboratory work and present the results to the lecturer.
Students who have missed classes without valid reasons and have not participated in current control activities are not admitted to the final semester control. In this case, a mark 'non-admission' is put in the exam record on the day of the exam.
Repeated taking of the exam of the discipline is appointed in case of accomplishing all types of educational, individual work stipulated by the working program of the academic discipline and is carried out according to the approved schedule of academic failure liquidation.

14 Policy of late task performance

Tasks and laboratory works submitted later are assessed with a lower grade. The grade is reduced by one point for each week of lateness.

15 Academic integrity policy

Participants in the educational process rely on the academic integrity principles. One should provide references to sources of information when using someone else's ideas, statements, data, as well as verified information.

16 Recommended sources of information

Primary literature:
1. Kovaliuk T.V. Algorithmization and programming. – K.: BHV Publishing Group, 2010. – 380 p.
2. Kovaliuk T.V. Basics of programming. – K.: BHV Publishing Group, 2005. – 384 p.
3. Wirth N. Algorithms + Data Structures = Programs. – M.: Mir, 1985. — 406 p.
4. Deitel H. M, Deitel P. J. C++ How to Program. – M.: CJSC "BINOM Publishing House", 200
5. Koenig A., Moo B. Accelerated C++: Practical Programming by Example. – M.: Williams Publishing House, 2002. — 384 p.
6. Lafore R. Object-Oriented Programming in C++. – St. Petersburg: Peter, 2006. — 926 p.
7. Prata S. C++ Primer Plus. – K.: DiaSoft Publishing House, 2005. – 1104 p.
8. Sedgewick R. Algorithms in C++: Fundamentals. – K.: DiaSoft Publishing House, 2001. – 688 p.
9. Schildt H. C++ A Beginner's Guide. – St. Petersburg: BHV-Petersburg, 2003. – 687 p.

Additional literature:
1. Aho A., Hopcroft J., Ullman J. Design and Analysis of Computer Algorithms. – M.: Mir, 1979. — 536 p.
2. Hilbert A. How to work with matrices. – M.: Statistics, 1981. – 160 p.
3. Lipsky V. Combinatorics for programmers. – M.: Mir, 1988. – 213 p.
4. Mainika E. Optimization algorithms on networks and graphs. – M.: Mir, 1981. – 434 p.
5. Okulov S. M. Programming in algorithms. – M.: BINOM. Laboratory of knowledge, 2002. — 341 p.
6. Novikov F. A. Discrete mathematics for programmers. – St. Petersburg: Peter, 2003. – 304 p.
7. Ivanov B. N. Discrete mathematics. Algorithms and programs. – M.: Laboratory of Basic Knowledge, 2002. — 228 p.
8. Porublev I. N., Stavrovsky A. B. Algorithms and programs. Solution of Olympiad problems. – M.: I.D. Williams LLC, 2007. – 480 c.
9. Meyers S. Effective C++: 55 Specific Ways to Improve Your Programs and Designs. – M.: DMK Press, 2006. – 300 p.
10. Sutter H. More Exceptional C++. – M.: Williams Publishing House, 2005. — 272 p.
11. Sutter H. Exceptional C++. – M.: Williams Publishing House, 2002. — 400 p.

Internet resources:
https://metanit.com/
https://sites.google.com/site/programuvannapaskal/osnoviprogramuvanna/mova-programuvanna-s
http://www.dut.edu.ua/ua/lib/1/category/1052/view/539

17 Tips on successful study during the course

Note: examine lecture materials and perform tasks and laboratory works synchronously with the curriculum. Thus, your abilities and insistence will be the key to success!