МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ

"ЗАТВЕРДЖУЮ"
Заступник керівника
(проректор з науково-педагогічної роботи)
_____
Микола АФАНАСЬЄВ

ПРОГРАМУВАННЯ

робоча програма навчальної дисципліни

Галузь знань	***12 Інформаційні технології***
Спеціальність	***121 Інженерія програмного забезпечення***
	122 Комп'ютерні науки
Освітній рівень	***перший (бакалаврський)***
Освітня програма	***Інженерія програмного забезпечення***
	Комп'ютерні науки

Статус дисципліни	***базова***
Мова викладання, навчання та оцінювання	***англійська***

Завідувач кафедри
кібербезпеки та
інформаційних технологій _____ *Сергій ЄВСЄЄВ*

Харків
2020

MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
SIMON KUZNETS KHARKIV NATIONAL UNIVERSITY OF ECONOMICS

PROGRAMMING

working program of the discipline

Field of knowledge	***12 Information technologies***
Speciality	***121 Software Engineering***
	122 Computer Science
Educational level	***first (bachelor's)***
Educational program	***Software engineering***
	Computer Science

Discipline status	***basic***
Language of instruction, teaching and assessment	***English***

Head of Department
cybersecurity and information technology _____ Serhii YEVSEIEV

Kharkiv
2020

APPROVED
at a meeting of the *Department of Cybersecurity and Information Technology*
Protocol № 2 dated 31.08.2020

Developer:
Milov O.V., Ph.D., Prof. of CIT Department.
Tkachev A.M., Ph.D., Assoc. Prof. of CIT Department.

Update and re-approval letter
working program of the discipline

Academic year	Date of the meeting of the department-developer of WP	Protocol number	Signature of the head of the department

Abstract of the discipline

The rapid development of information technology (IT) in the modern world facilitates the application of computer systems and solutions in any field of human activity. Software development based on the application of object-oriented approach to programming allows you to develop complex software solutions in less time and effectively coordinate management processes for development teams. The current state of development of programming tools and related tools allows us to identify the C language, which is controlled, and actually implements the latest approaches to programming complex tasks.

In today's global information space, experts in a particular subject area should know the main trends in the development of new programming technologies, navigate in services that provide cloud computing (Cloud Computing) to effectively develop software products. The course provides: professional acquaintance with the features of algorithmic approach to programming based on Microsoft technologies.

The object of study of the discipline are the processes of software development of modern information and communication systems.

The subject of the discipline is the C programming language, algorithmic approach to the development of complex systems and programming tools.

The purpose of the discipline is to master the theoretical foundations and the formation of practical skills in future bachelors in programming using the tools and methods.

The results of the study of the discipline are the acquisition of practical skills in the use of software development tools and mastering effective means of limiting the risks of creating software.

Characteristics of the discipline

Course	**1, 2**
Semester	**1, 2**
Number of ECTS credits	**10**
Form of final control	**Exam**

Structural and logical scheme of studying the discipline

Prerequisites	Postrequisites
Computer science according to the school program	Object-oriented programming
	Data bases
	Distributed and parallel computation
	WEB-technologies and WEB-design
	Software development and testing technologies

Competences and learning outcomes in the discipline

Competences	Learning outcomes
Ability to abstract thinking, analysis and synthesis. Ability to use information and communication technologies to search for new information, program application software in the professional field, the application of object-oriented approach to software development, the use of computer software systems and their optimization.	Know the basic processes, phases and iterations of the software life cycle. Know and apply in practice the fundamental concepts, paradigms and basic principles of operation of language, instrumental and computational software engineering. Know and apply relevant mathematical concepts, methods of domain, system and object-oriented analysis and mathematical modeling for software development.

Ability to identify, classify and formulate software requirements. Ability to solve specialized problems and practical problems in using an object-oriented approach to software development.	Analyze, purposefully search for and select the necessary information and reference resources and knowledge to solve professional problems, taking into account modern advances in science and technology. Know and be able to use methods and tools for collecting, formulating and analyzing software requirements. Apply effective approaches to software design in practice.

Curriculum of the discipline

(1 semester)

Content module 1. Mathematical foundations of programming

Topic 1. *Mathematical foundations of computer science.*
Topic 2. *Information units and digital systems*.
Topic 3. *Algorithms*.
Topic 4. *Data types.*

Content module 2. Data calculation

Topic 5. *Data search algorithms.*
Topic 6. *Data sorting.*
Topic 7. *Linear data structures.*
Topic 8. *Data hashing.*

(2 semester)

Content module 3. Introduction to programming

Topic 9. *Programming Concepts.*
Topic 10. *Software product design*.
Topic 11. *Software product architecture*.
Topic 12. *Development technologies.*

Content module 4. Developer tools

Topic 13. *Operating Systems.*
Topic 14. *Programming Languages.*
Topic 15. *Developer tools.*
Topic 16. *Business-process automation.*

The list of laboratory classes, as well as questions and tasks for independent work is given in the table "Rating-plan of the discipline".

Teaching and learning methods

In the course of teaching the discipline the teacher uses explanatory-illustrative (information-receptive) and reproductive teaching methods. Problem-based lectures, presentations, conversations, individual and group mini-projects are used as teaching methods that are aimed at activating and stimulating the educational and cognitive activities of applicants.

The procedure for evaluating learning outcomes

The system of assessment of formed competencies in students takes into account the types of classes, which in accordance with the curriculum of the discipline include lectures and laboratory classes, as well as independent work. Assessment of the formed competencies of students is carried out according to the accumulative 100-point system. Control measures include:

1) current control, which is carried out during the semester during lectures and laboratory classes and is estimated by the amount of points scored (maximum amount - 100 points; the minimum amount that allows a student to set off - 60 points);

2) final / semester control, which is conducted in the form of a exam, in accordance with the schedule of the educational process for 1 and 2 semesters.

The procedure for the current assessment of students' knowledge.

Assessment of student knowledge during lectures and laboratory classes is carried out according to the following criteria:

- know the basics of algorithmic approach;
- know the basic software constructs of the C language;
- apply the programming paradigm: abstraction, encapsulation, imitation and polymorphism;
- develop UML class diagrams;
- handle exceptions in the program in C.
- use software design templates;
- apply the SOLID principles used for the design and development of software systems.

The discipline provides the following methods of current formative assessment: interviews and oral comments of the teacher on his results, instructions of teachers in the process of laboratory tasks, the formation of self-assessment skills and discussion of completed laboratory tasks, control of individual performance.

All work must be done independently in order to develop a creative approach to solving problems.

Lectures:

1st semester - the maximum number of points is 10 (work on lectures);

2nd semester - the maximum number of points is 4 (work on lectures);

Laboratory classes:

1st semester - the maximum number of points is 90 (laboratory work - 10, defense of laboratory work - 35, control work - 45), and the minimum - 60;

2nd semester - the maximum number of points is 56 (laboratory work - 4, defense of laboratory work - 24, control work - 28), and the minimum - 60.

Independent work in 1 and 2 semesters: consists of time that the applicant spends on preparation for laboratory work and preparation for express surveys of lectures and tests for laboratory work of the discipline, in the technological map points for this type of work are not allocated.

Final control in the first semester: is carried out taking into account the exam.

A student should be considered certified if the sum of points obtained from the results of the final / semester performance test is equal to or exceeds 60.

Final control in the second semester: is carried out taking into account the exam.

The examination ticket covers the program of the discipline and provides for the determination of the level of knowledge and the degree of mastery of competencies by students.

Each exam ticket consists of 3 practical situations (one stereotypical, one diagnostic and one heuristic task), which involve solving typical professional tasks in the workplace and allow to diagnose the level of theoretical training of the student and his level of competence in the discipline. Evaluation of each task of the examination ticket is as follows: the first task is 20 test tasks of the closed form, its performance is estimated by 20 points; the second task is devoted to the development of program code for the task, its implementation is evaluated by 10 points; the third task - debugging the program code, its execution is estimated by 10 points.

The result of the semester exam is evaluated in points (maximum number - 40 points, minimum number of credits - 25 points) and is affixed in the appropriate column of the examination "Information of performance".

A student should be considered certified if the sum of points obtained from the final / semester test is equal to or exceeds 60. The minimum possible number of points for current and modular control during the semester is 35 and the minimum possible number of points scored in the exam is 25.

The final grade in the discipline is calculated taking into account the points obtained during the current control of the accumulative system. The total result in points for the semester is: "60 or more points - credited", "59 or less points - not credited" and is entered in the test "Statement of performance" of the discipline.

The final grade is set according to the scale given in the table "Assessment scale: national and ECTS".

Forms of assessment and distribution of points are given in the table "Rating-plan of the discipline".

Assessment scale: national and ECTS

The sum of points for all types of educational activities	Score ЄКТС	Score on a national scale	
		for exam, course project (work), practice	For credit
90 – 100	A	excellent	credited
82 – 89	B	fine	
74 – 81	C		
64 – 73	D	satisfactorily	
60 – 63	E		
35 – 59	FX	unsatisfactorily	Not credited

Rating plan of the discipline
1st semester

Topic	Forms and types of education		Forms of essesment	Max points
Topic 1	***Classroom work***			
	Lecture	*Lecture* 1. *"Mathematical foundations of computer science"*	Lecture	2
	Laboratory lesson	*Laboratory work 1. Operation research under the sets.*	Laboratory lesson	6
	Individual work			
	Questions and tasks for self-study	*Search, selection and review of literary sources on a given topic. Preparation for laboratory work. Execution of laboratory tasks. Laboratory lesson*		
Topic 2.	***Classroom work***			
	Lecture	Lecture 2. *"Information units and digital systems"*	Lecture	1
	Laboratory lesson	*Laboratory work 2. Conversation the number from one to another number systems.*	Laboratory lesson	3
	Individual work			
	Questions and tasks for self-study	*Search, selection and review of literary sources on a given topic. Preparation for laboratory work. Execution of laboratory tasks. Laboratory lesson*		
Topic 3	***Classroom work***			
	Lecture	Lecture *"Algorithms"*	Lecture	1
	Laboratory lesson	*Laboratory work 3. Algorithm research.*	Laboratory lesson	3
	Individual work			

	Questions and tasks for self-study	*Search, selection and review of literary sources on a given topic. Preparation for laboratory work. Execution of laboratory tasks. Laboratory lesson*		
Topic 4	***Classroom work***			
	Lecture	Lecture *"Data types"*	Lecture	2
	Laboratory lesson	*Laboratory work 4. Data types research.*	Laboratory lesson	6
			Control test 1	6
	Individual work			
	Questions and tasks for self-study	*Search, selection and review of literary sources on a given topic. Preparation for laboratory work. Execution of laboratory tasks. Laboratory lesson*		
Topic 5	***Classroom work***			
	Lecture	Lecture *"Data search algorithms"*	Lecture	2
	Laboratory lesson	*Laboratory work 5. Search algorithms research.*	Laboratory lesson	6
	Individual work			
	Questions and tasks for self-study	*Search, selection and review of literary sources on a given topic. Preparation for laboratory work. Execution of laboratory tasks. Laboratory lesson*		
Topic 6	***Classroom work***			
	Lecture	Lecture *"Data sorting"*	Lecture	1
	Laboratory lesson	*Laboratory work 6. Data Sorting research.*	Laboratory lesson	3
	Individual work			
	Questions and tasks for self-study	*Search, selection and review of literary sources on a given topic. Preparation for laboratory work. Execution of laboratory tasks. Laboratory lesson.*		
Topic 7	***Classroom work***			
	Lecture	Lecture *"Linear data structures"*	Lecture	2
	Laboratory lesson	*Laboratory work 6. Linear structures.*	Laboratory lesson	6
	Individual work			
	Questions and tasks for self-study	*Search, selection and review of literary sources on a given topic. Preparation for laboratory work. Execution of laboratory tasks. Laboratory lesson. Preparation to the final test.*		
Topic 8	***Classroom work***			
	Lecture	Lecture *"Data hashing"*	Lecture	1

	Laboratory lesson	*Laboratory work 7. Hashing usage*	Laboratory lesson	3
			Control test 2	6
	Individual work			
	Questions and tasks for self-study	*Search, selection and review of literary sources on a given topic. Preparation for laboratory work. Execution of laboratory tasks. Laboratory lesson. Preparation to the final test.*		
Exam				40

Rating plan of the discipline
2nd semester

Topic	Форми та види навчання		Форми оцінювання	Max бал
Topic 9	***Classroom work***			
	Lecture	Lecture *"Programming Concepts"*	Lecture	2
	Laboratory lesson	*Laboratory work 1. Programming concepts research*	Laboratory lesson	6
	Individual work			
	Questions and tasks for self-study	*Search, selection and review of literary sources on a given topic. Preparation for laboratory work. Execution of laboratory tasks. Laboratory lesson*		
Topic 10.	***Classroom work***			
	Lecture	Lecture *"Software product design"*	Lecture	1
	Laboratory lesson	*Laboratory work 2. Introduction to design steps*	Laboratory lesson	3
	Individual work			
	Questions and tasks for self-study	*Search, selection and review of literary sources on a given topic. Preparation for laboratory work. Execution of laboratory tasks. Laboratory lesson*		
Topic 11	***Classroom work***			
	Lecture	Lecture *"Software product architecture"*	Lecture	1
	Laboratory lesson	*Laboratory work 2. Typical construction*	Laboratory lesson	3
	Individual work			
	Questions and tasks for self-study	*Search, selection and review of literary sources on a given topic. Preparation for laboratory work. Execution of laboratory tasks. Laboratory lesson*		
і с	***Classroom work***			

	Lecture	Lecture *"Development technologies"*	Lecture	2
	Laboratory lesson	*Laboratory work 3. Introduction to development of program technologies*	Laboratory lesson	6
			Control test 3	6
	Individual work			
	Questions and tasks for self-study	*Search, selection and review of literary sources on a given topic. Preparation for laboratory work. Execution of laboratory tasks. Laboratory lesson*		
Topic 13	***Classroom work***			
	Lecture	Lecture *"Operating Systems"*	Lecture	2
	Laboratory lesson	*Laboratory work 5. Introduction to Operating systems*	Laboratory lesson	6
	Individual work			
	Questions and tasks for self-study	*Search, selection and review of literary sources on a given topic. Preparation for laboratory work. Execution of laboratory tasks. Laboratory lesson*		
Topic 14	***Classroom work***			
	Lecture	Lecture *"Programming Languages"*	Lecture	1
	Laboratory lesson	*Laboratory work 6. Introduction to programming languages*	Laboratory lesson	3
	Individual work			
	Questions and tasks for self-study	*Search, selection and review of literary sources on a given topic. Preparation for laboratory work. Execution of laboratory tasks. Laboratory lesson. Preparation to the final test.*		
Topic 15	***Classroom work***			
	Lecture	Lecture *" Developer tools"*	Lecture	2
	Laboratory lesson	*Laboratory work 7. Developer tools*	Laboratory lesson	6
	Individual work			
	Questions and tasks for self-study	*Search, selection and review of literary sources on a given topic. Preparation for laboratory work. Execution of laboratory tasks. Laboratory lesson. Preparation to the final test.*		
Topic 16	***Classroom work***			
	Lecture	Lecture *"Business-process automation"*	Lecture	1

	Laboratory lesson	*Laboratory work 8. Business-process automation research.*	Laboratory lesson	3
			Control test 4	6
	Individual work			
	Questions and tasks for self-study	*Search, selection and review of literary sources on a given topic. Preparation for laboratory work. Execution of laboratory tasks. Laboratory lesson. Preparation to the final test.*		
Exam				40

Recommended Books

Basic

1. Object-oriented programming: a synopsis of lectures for students in the field of training "Computer Science" of all forms of education / Yu. E. Parfenov, V. M. Fedorchenko, M. Yu. Losev, OV Shcherbakov.– Kharkiv: Ed. KhNEU, 2010.– 312p.
2. Methodical recommendations for performance of laboratory works on discipline "Object-oriented programming" for students of a direction of preparation "Computer sciences" of all forms of training. Part 1 / Comp. u. E. Parfenov, V. M. Fedorchenko, M. Yu. Losev, OV Shcherbakov - H .: Ed. KhNEU, 2008. - 72 p.
3. Object-oriented programming. Part 1. Fundamentals of object-oriented programming in C # .: Tutorial. / D.V. Nastenko, AB Nesterko. - K .: NTUU "KPI", 2016. - 76p. [Electronic resource]. - Access mode: http://ela.kpi.ua/bitstream/123456789/16671/1/OOP_manual.pdf
4. Object-oriented programming. Laboratory workshop: textbook / B.I. Boyko, L.L. Omelchuk, NG Rusina - K .: 2016. - 90 p. [Electronic resource]. - Access mode: http://csc.knu.ua/media/filer_public/4a/35/4a3533cd-4ec7-45f3-85d2-4edaafdf1b82/oop_2016.pdf
5. C# Notes for Professionals book [Електронний ресурс]. – Режим доступу : https://books.goalkicker.com/CSharpBook/
6. Fundamentals of Computer Programming with C#. Authors: Svetlin Nakov and Team. Publisher: Faber, Veliko Tarnovo, Bulgaria, 2013, Pages: 1122 [Електронний ресурс]. – Режим доступу : https://introprogramming.info/english-intro-csharp-book/
7. The Free Book + Video Course "Programming Basics with C#" [Електронний ресурс]. – Режим доступу : https://csharp-book.softuni.org/

Additional

8. Weisfeld M. Object-Oriented Thinking - 2014, 304 pp., ISBN: 978-5-496-00793-1, Peter.
9. Herbert Schildt. C # 4.0: The Complete Guide - 1056 pp., ISBN 978-5-8459-1684-6, hardcover; 2015, Williams.
10. Richter D. CLR via C #. Programming on Microsoft .NET Framework 4.5 in C # - 2016, 896 pages, ISBN: 978-5-496-00433-6, Peter.
11. Adam Fremen. ASP.NET Core MVC with examples in C # for professionals // Williams - 2017 - 992 p.
12. Object-oriented analysis and design with examples of applications (UML 2). Third edition. Grady Booch, Robert A. Maximchuk, Michael W. Engle, Bobby J. Young, Jim Conallen, Kelly A. Houston - 720 pages, ISBN 978-5-8459-1401-9, hardcover; 2010, Williams ..
13. Laforêt R. Object-oriented programming in C ++. Classics Computer Science - 2016, 928 pp., ISBN: 978-5-496-00353-7, Peter.

Information resources.

14. Section on C # programming language and .NET platform on the METANIT.COM website [Electronic resource]. - Access mode: https://metanit.com/sharp/

15. Object Oriented Programming in C #. [Electronic resource] Platform for mass open online courses edX. Developer: Microsoft. - Access mode: https://www.edx.org/course/object-oriented-programming-in-c-3

16. C # - Channel 9 programming language [Electronic resource]. - Access mode: https://channel9.msdn.com/Series/C-Development-Russian

17. C # Guide [Electronic resource]. - Access mode: https://docs.microsoft.com/en-us/dotnet/csharp/.

18. .NET Core Guide [Electronic resource]. - Access mode: https://docs.microsoft.com/en-us/dotnet/core/

19. .NET Tutorial - Hello World in 10 minutes [Electronic resource]. - Access mode: https://dotnet.microsoft.com/learn/dotnet/hello-world-tutorial/intro

20. Site of personal educational systems of S. Kuznets KhNEU in the discipline "Programming" https://pns.hneu.edu.ua/enrol/index.php?id=7009.