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

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

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ ІМЕНІ СЕМЕНА КУЗНЕЦЯ
№02071211

ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

робоча програма навчальної дисципліни

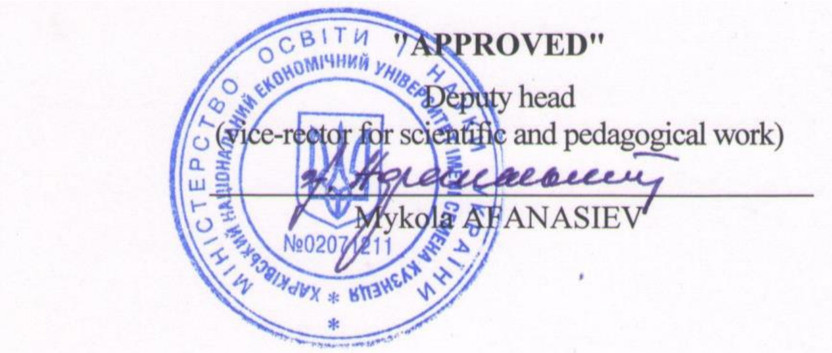
Галузь знань	*12 Інформаційні технології*
Спеціальність	*121 Інженерія програмного забезпечення*
Освітній рівень	*перший (бакалаврський)*
Освітня програма	*Інженерія програмного забезпечення*

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

Завідувач кафедри
кібербезпеки та
інформаційних технологій ______________ *Сергій ЄВСЕЄВ*

Харків
2020

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



"APPROVED"
Deputy head
(vice-rector for scientific and pedagogical work)
____
Mykola AFANASIEV

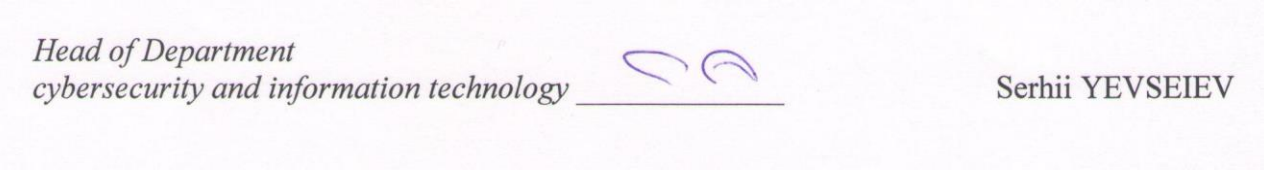
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ ІМЕНІ СЕМЕНА КУЗНЕЦЯ
№02071211

OBJECT-ORIENTED PROGRAMMING

syllabus of the discipline

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

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:
Shmatko OV, 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, such as Codenvy. The course provides: professional acquaintance with the features of object-oriented approach to programming based on Microsoft .Net Core 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, object-oriented 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 of object-oriented approach.

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

Cource	**2**
Semester	**3, 4**
Number of ECTS credits	**11**
Form of final control	**Credit, exam**

Structural and logical scheme of studying the discipline

Prerequisites	**Postrequisites**
Basics of programming	Development and analysis of algorithms
Higher mathematics	

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.	Analyze, purposefully search for and select the necessary information and reference resources and knowledge to solve

Ability to solve specialized problems and practical problems in using an object-oriented approach to software development.	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

Content module 1. Fundamentals of programming of modern computer systems using object-oriented approach

Topic 1. Introduction to .NET Framework and .NET Core technologies

Topic 2. Basics of object-oriented C # programming language

Topic 3. Fundamentals of object-oriented programming

Content module 2. Features and examples of object-oriented approach

Topic 4. Object-oriented design

Topic 5. Features of the .NET Framework and .NET Core runtime

Content module 3. Templates and principles of Software design

Topic 6. Software design templates

Topic 7. SOLID principles used for the design and development of software systems

Content module 4. Features and examples of application of object-oriented approach in solving problems of web systems development

Topic 8. Interaction with databases

Topic 9. Development of web-based systems

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 test, in accordance with the schedule of the educational process for 3 and 4 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 object-oriented 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:

3rd semester - the maximum number of points is 10 (work on lectures);

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

Laboratory classes:

3rd semester - the maximum number of points is 90 (laboratory work - 10, defense of laboratory work - 35, control work - 45), and the minimum - 60;

4th 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 3 and 4 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 third semester: is carried out taking into account the points obtained during the semester.

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 fourth 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
3rd semester

Topic	Forms and types of education		Forms of evaluation	Max points
Topic 1	***Classroom work***			
	Lecture	*Lecture "Introduction to .NET Framework and .NET Core"*	Lecture	2
	Laboratory lesson	*Laboratory work 1. Fundamentals of object-oriented approach. First C # program*	Laboratory lesson	2
	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		
Topic 2	***Classroom work***			
	Lecture	*Lecture "Fundamentals of object-oriented programming language C #"*	Lecture	2
	Laboratory lesson	Laboratory work 2. Basic program constructions of C # language.	Laboratory lesson	2
			Answer of laboratory work 1, 2	14
	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		
Topic 3	***Classroom work***			
	Lecture	Лекція *"Основи об'єктно-орієнтованого програмування"Lecture "Fundamentals of Object Oriented Programming"*	Lecture	2

	Laboratory lesson	*Laboratory work 3. Application of programming paradigms: abstraction, encapsulation, imitation and polymorphism*	Laboratory lesson	2
			Control work 1	22
	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		
Topic 4	***Classroom work***			
	Lecture	*Lecture "Object-oriented design"*	Lecture	2
	Laboratory lesson	*Laboratory work 4. Development of UML-diagram of classes.*	Laboratory lesson	2
			Answer of laboratory work 3, 4	14
	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		
Topic 5	***Classroom work***			
	Lecture	Лекція *"Особливості середовища виконання програми .NET Framework та .NET Core"Lecture "Features of the .NET Framework and .NET Core runtime"*	Lecture	2
	Laboratory lesson	*Laboratory work 5. Processing of exceptional situations in the program on C #*	Laboratory lesson	2
			Answer of laboratory work 5	7
			Control work 2	23
	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		

Rating plan of the discipline
4 semester

Topic	**Forms and types of education**	**Forms of evaluation**		**Max points**
Topic 6	***Classroom work***			
	Lecture	*Lecture "Software Design Templates"*	Lecture	1
	Laboratory lesson	*Laboratory work 6. Familiarity with*	Laboratory	1

		software design templates	lesson	
	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		
Topic 7	***Classroom work***			
	Lecture	*Lecture "SOLID principles used for design and development of software systems"*	Laboratory lesson	1
	Laboratory lesson	*Laboratory work 7. Application of SOLID principles used for design and development of software systems*	Laboratory lesson	1
			Answer of laboratory work 6	6
			Control work 3	14
	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		
Topic 8	***Classroom work***			
	Lecture	*Lecture "Interaction with databases"*	Laboratory lesson	1
	Laboratory lesson	*Laboratory work 8. Development using Entity Framework*	Laboratory lesson	1
			Answer of laboratory work 7	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		
Topic 9	***Classroom work***			
	Lecture	*Lecture "Development of web-based systems"*	Laboratory lesson	1
	Laboratory lesson	*Laboratory work 9. Application of ASP.NET MVC Framework technology.*	Laboratory lesson	1
			Answer of laboratory work 8, 9	12
			Control work 4	14
	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		
Final 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 "Object-Oriented Programming" https://pns.hneu.edu.ua/enrol/index.php?id=5528.