**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ**

**"ЗАТВЕРДЖУЮ"**

Заступник керівника
(проректор з науково-педагогічної роботи)

М.В. 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** |

| | |
|---|---|
| Вид дисципліни | **базова** |
| Мова викладання, навчання та оцінювання | **англійська** |

Завідувач кафедри *кібербезпеки*
*та інформаційних технологій*

Євсеєв С.П.

Харків
ХНЕУ ім. С. Кузнеця
2019

MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

SIMON KUZNETS KHARKIV NATIONAL UNIVERSITY OF ECONOMICS

# OBJECT-ORIENTED PROGRAMMING

## Syllabus of the academic discipline

| | |
|---|---|
| Area of expertise | **All** |
| Speciality | **121 Program engineering** |
| Grade level | **The first (Bachelor's level)** |
| Academic Program | **Program engineering** |

| | |
|---|---|
| Type of discipline | **basic** |
| Teaching, studying and evaluating language | **English** |

Kharkiv
S. Kuznets KhNUE

2019

APPRUVED
At the session of the Cybersecurity and Information Technology Department
Protocol № 1 from 26.08. 2019.

Drafters:
Volodymyr ALEKSIYEV, Doctor of Sciences, Professor of Cybersecurity and Information Technology Department

**The list of renewal and re-approval of academic discipline program**

| Academic year | Data of the session of the Department – Drafter of SDAD | Protocol Number | Signature of Head of Department |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

## 1. Introduction

**Annotation for the academic discipline:**

The rapid development of information technology (IT) in the modern world contributes to the integration of computer systems and solutions in every area of human activity. Software development with an object-oriented programming approach enables the development of highly sophisticated software solutions in a shorter time and effectively aligns management processes for development teams. The current state of development of programming tools makes it possible to distinguish C#, which is manageable, and implements the latest programming approaches for complex tasks.

In today's global information space, professionals in a subject area should be aware of the main trends in the development of new software technologies, be guided by Cloud Computing services to perform effective software development, such as Codenvy. The course provides: a professional introduction to the features of an object-oriented approach to programming using Microsoft .Net Core technologies.

The discipline of Object-Oriented Programming is a basic discipline. It is taught in the first and second semesters of undergraduate studies in the amount of 150 hours. (5 ECTS credits) per semester. The course has four content modules and five test robots. The discipline ends with an exam.

The subject of study of the discipline are the processes of software development of modern Information and Communication Systems.

The subject of the course is C# programming language, object-oriented approach to complex systems development, and programming tools.

**The Purpose of the academic discipline:**

is the acquisition of theoretical foundations and the formation of practical skills in future bachelors in programming with the use of tools and methods of object-oriented approach.

The main tasks of studying the discipline are:

- learning the principles and rules of application of object-oriented approach in the subject area;
- learning the basics of functioning of .Net Core technologies and to define peculiarities of programming of applications of modern operating systems, packages of application programs and cloud services;
- acquire the ability to use software tools for software development;
- master effective risk mitigation tools to create a software program.

| | | | |
|---|---|---|---|
| Course | | **2** | **2** |
| Semester | | **1 (3)** | **2 (4)** |
| Number of ECTS credits | | **5** | **5** |
| Audit lessons | **lectures** | **32** | **32** |
| | **laboratory** | **32** | **32** |
| Independent work | | **86** | **86** |
| Form of final control | | **Credit** | **Exam** |

## 2. Competence and results of studying a discipline:

| Competence | Learning outcomes |
|---|---|
| Ability to think abstractly, analyze and synthesize. Ability to use information and communication technologies to find new information, to program application software in the professional field, to use object-oriented approach to software creation, to use computer software systems and their optimization. | Know the basic processes, phases, and iterations of the software lifecycle. To know and put into practice the fundamental concepts, paradigms and basic principles of the functioning of languages, tools and computational tools of software engineering. Know and apply relevant mathematical concepts, domain, system and object-oriented methods of analysis and mathematical modeling to software development. |
| Ability to identify, classify and formulate software requirements. Ability to solve specialized problems and practical problems with the use of object-oriented approach for software creation. | Analyze, purposefully seek and select the necessary information and reference resources and knowledge necessary for solving professional tasks, taking into account the modern achievements of science and technology. To know and be able to use methods and tools for collecting, formulating and analyzing software requirements. Implement effective software design approaches in practice. |

## 3. Program of the discipline

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

**Topic 1. Introduction to .NET Framework and .NET Core Technologies**

1.1. A brief history of the development of computer hardware and operating systems.

1.2. Fundamentals of structural programming.

1.3. Features of the .NET Framework and .NET Core technologies.

1.4. The basics of object-oriented approach. The first program in C#.

**Topic 2. The basics of C# object-oriented programming language**

3.1. Basic software constructions of C# language.

3.2. Declare and Call methods in C#.

3.3. Creating new types in C#. Collections and Generics.

**Topic 3. The basics of object-oriented programming**

3.1. Abstraction. The concept of class and object.

3.2. Encapsulation of data and methods. Overloading operations.

3.3. Inheritance and polymorphism.

## Content module 2.
## Features and examples of application which is based an object-oriented approach

### Topic 4. Object-oriented design

4.1. UML class diagrams.

4.2. Abstract classes. Interfaces and virtual methods.

4.3. Class reuse. The concept of association. The relation of composition and aggregation as types of association.

### Topic 5. Features of the .NET Framework and .NET Core Runtime

5.1. Exception handling.

5.2. The Garbage Collector works.

5.3. Asynchronous operations.

## Content module 3.
## Design templates and principles of Software Design

### Topic 6. Software Design Templates

6.1. Generating Patterns (Creational). Creating new objects on the system.

6.2. Structural patterns. Solve system-class and object-based layout tasks.

6.3. Behavioral Patterns. Allocation of responsibilities between objects in the system.

### Topic 7. SOLID principles used to design and develop software systems

7.1. Single responsibility principle.

7.2. Open / closed principle.

7.3. Liskov substitution principle.

7.4. Interface segregation principle.

7.5. Dependency inversion principle.

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

### Topic 8. Database interaction

8.1. ORM (Object-relational mapping) technology.

8.2. Introduction to the Entity Framework.

### Topic 9. Development of web-oriented systems

9.1. Model-view-controller template.

9.2. Practical application of ASP.NET MVC Framework technology

## Off-class work of students

Mastering the practical skills and learn the core concepts of object-oriented programming (OOP) using the C# language based on the online course “Object Oriented Programming in C#” on EdX (https://www.edx.org/course/object-oriented-programming-in-c). Learn how to improve the performance of source code with advanced techniques based on the online course “Writing Professional Code - Advanced Topics” on EdX (https://www.edx.org/ course/ writing-professional-code-advanced-topics).

## Laboratory Workshop

**Laboratory work 1.**
Fundamentals of object-oriented approach. The first program in C#.

**Laboratory work 2.**
Basic software constructions of C# language.

**Laboratory work 3.**
Application of programming paradigms: abstraction, encapsulation, inheritance and polymorphism.

**Laboratory work 4.**
Development of UML-class diagram.

**Laboratory work 5.**
C# program exception handling.

**Laboratory work 6.**
Introduction to software design templates.

**Laboratory work 7.**
Application of the SOLID principles used for design and development of software systems.

**Laboratory work 8.**
Development using Entity Framework.

**Laboratory work 9.**
Application of ASP.NET MVC Framework technology.

## 4. The procedure of evaluation of the learning results

The system of evaluation of the developed competencies of students considers the types of occupations, which according to the curriculum program include lectures, seminars, classes, as well as independent work. Assessment of the developed competencies in students is carried out using a 100-point accumulation system. In accordance with the Provisional Regulations "On the Procedure for Assessing the Results of Students' Learning Based on the Accumulated Bulletin-Rating System" S. Kuznets KhNEU, control measures include:

– current control over the semester during lectures and laboratory classes and is estimated by the sum of the points scored (the maximum amount is 60 points; the minimum amount that allows the student to take the exam - 35 points);

– modular control as an intermediate testing on the initiative of the teacher, considering the current control over the relevant content module and aims to integrate the evaluation of the student's learning outcomes after studying the material from the logically completed part of the discipline - content module;

– final / semester control, conducted in the form of a credit, according to the schedule of the educational process.

The procedure for carrying out the current assessment of students' knowledge. Assessment of students' knowledge during lecture and laboratory classes and fulfillment of individual tasks is carried out according to the following criteria: understanding of the object-oriented approach to software development, development of programs in the C# language with a newest instrumental toolset. Students should be guided in the best practices of programming and using well-known patterns. Modular control tests and exam questions contain practical issues that require knowledge and understanding of the fundamentals of object-oriented programming. Relevant theoretical questions are related to understanding and applying the best recommendations for writing modern programs.

The general criteria for evaluating independent work of students are: the depth and strength of knowledge, the level of thinking, the ability to systematize knowledge on specific topics, the ability to make conclusions, the ability to find a solution of problems in uncertain situations using an object-oriented approach to the development of complex software systems.

**The final control** of the knowledge and competences of students in the discipline is based on a score that is considered to be successful if the student scored 60 points or more during the semester.

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

**Distribution of points by weeks**

| Topics of the content module | | | Lecture classes | Laboratory classes | **Testing** | **Total** |
|---|---|---|---|---|---|---|
| **Content module 1** | **Topic 1** | 1 Week | 0,5 | 1 | | **1,5** |
| | | 2 Week | 0,5 | 1 | | **1,5** |
| | **Topic 2** | 3 Week | 0,5 | 1 | | **1,5** |
| | | 4 Week | 0,5 | 11 | | **11,5** |
| | | 5 Week | 0,5 | 1 | | **1,5** |
| | **Topic 3** | 6 Week | 0,5 | 1 | | **1,5** |
| | | 7 Week | 0,5 | 1 | | **1,5** |
| | | 8 Week | 0,5 | 11 | | **11,5** |
| **Content module 2** | **Topic 4** | 9 Week | 0,5 | 1 | 8 | **8,5** |
| | | 10 Week | 0,5 | 1 | | **1,5** |
| | | 11 Week | 0,5 | 1 | | **1,5** |
| | | 12 Week | 0,5 | 11 | | **11,5** |
| | | 13 Week | 0,5 | 1 | 8 | **8,5** |
| | **Topic 5** | 14 Week | 0,5 | 1 | | **1,5** |
| | | 15 Week | 0,5 | 1 | | **1,5** |
| | | 16 Week | 0,5 | 11 | | **11,5** |
| | | 17 Week | | | 10 | **10** |
| | | *Test* | | | 10 | **10** |
| **Усього** | | | **8** | **56** | **36** | **100** |
| **Content module 3** | **Topic 6** | 1 Week | 0,5 | 0,5 | | **1,5** |
| | | 2 Week | 0,5 | 0,5 | | **1,5** |
| | | 3 Week | 0,5 | 0,5 | | **1,5** |
| | | 4 Week | 0,5 | 10,5 | | **11,5** |
| | **Topic 7** | 5 Week | 0,5 | 0,5 | | **1,5** |
| | | 6 Week | 0,5 | 0,5 | | **1,5** |
| | | 7 Week | 0,5 | 0,5 | | **1,5** |
| | | 8 Week | 0,5 | 10,5 | | **11,5** |
| **Content module 4** | **Topic 8** | 9 Week | 0,5 | 0,5 | 4 | **8,5** |
| | | 10 Week | 0,5 | 0,5 | | **1,5** |
| | | 11 Week | 0,5 | 0,5 | | **1,5** |
| | | 12 Week | 0,5 | 10,5 | | **11,5** |
| | | 13 Week | 0,5 | 0,5 | | **8,5** |
| | **Topic 9** | 14 Week | 0,5 | 0,5 | | **1,5** |
| | | 15 Week | 0,5 | 0,5 | | **1,5** |
| | | 16 Week | 0,5 | 10,5 | | **11,5** |
| | | 17 Week | | | | |
| | | *Exam* | | | 40 | **40** |
| **Total** | | | **8** | **48** | **44** | **100** |

The final score in the discipline is calculated on the basis of the points obtained during the exam and the points obtained during the current control over the accumulation system. The total score in the points for a semester is: "60 and more points -" enrolled "," 59 and less points - not taken into account "and entered in the" Record of success "of the academic discipline.

**Scale: national and ECTS**

| The amount of points for all types of educational activities | Rating ECTS | Score on a national scale | |
|---|---|---|---|
| | | for exam, course project (work), practice | for the offset |
| 90 – 100 | A | perfectly | Accepted |
| 82 – 89 | B | well | |
| 74 – 81 | C | | |
| 64 – 73 | D | satisfactorily | |
| 60 – 63 | E | | |
| 35 – 59 | FX | unsatisfactorily | not accepted |
| 1 – 34 | F | | |

## 5. Recommended Books

### *5.1 Main*

1. The Object-Oriented Thought Process, Fourth Edition / Matt Weisfeld. – Addison-Wesley, 2013.– 336 p.

2. Hands-On Software Architecture with C# 8 and .NET Core 3: Architecting software solutions using microservices, DevOps, and design patterns for Azure Cloud / Gabriel Baptista, Francesco Abbruzzese. – Packt Publishing Ltd, 2019. - 598 p.

3. Dive Into DESIGN PATTERNS / Alexander Shvets. – Refactoring.Guru, 2019 p. [Electronic resource]. – Acces mode : https://refactoring.guru/design-patterns/book

4. Object-Oriented Programming in C++ / Robert Lafore. - Pearson Education, 1997 p. – 1040 p.

5. C# Notes for Professionals book [Electronic resource]. – Acces mode : 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 [Electronic resource]. – Acces mode : https://introprogramming.info/english-intro-csharp-book/

7. The Free Book + Video Course "Programming Basics with C#" [Electronic resource]. – Acces mode : https://csharp-book.softuni.org/

### *5.2 Additional*

8. C# 4.0 The Complete Reference / Herbert Schildt, McGraw Hill Professional, 2010. - 976 p.

9. Pro C# 5.0 and the .NET 4.5 Framework / Andrew Troelsen, Apress, 2012. – 1560 p.

10. CLR via C# / Jeffrey Richter, Microsoft Press; 4 edition, 2012. – 896 p.

11. C# in Depth 4th Edition / Jon Skeet, Manning Publications; 4th edition. – 2019. – 528 p.

12. UML Distilled: A Brief Guide to the Standard Object Modeling Language / Martin Fowler, Addison-Wesley Professional, 2004. - 175 p.

### *5.3 Information resources of the Internet*

13. Object Oriented Programming in C#. [Electronic resource]. edX. / Microsoft. – Acces mode : https://www.edx.org/course/object-oriented-programming-in-c-3

14. DEV204.2x -Object Oriented Programming in C# - Microsoft [Electronic resource]. – Acces mode : https://github.com/msmithnova/DEV204.2x

15. .NET Conf 2017. C# Part 1 - Introduction to C#. / Bill Wagner [Electronic resource]. – Acces mode : https://channel9.msdn.com/Events/dotnetConf/2017/T313

16. C# Fundamentals for Absolute Beginners. - Channel 9 [Electronic resource]. – Acces mode : https://channel9.msdn.com/Series/CSharp-Fundamentals-for-Absolute-Beginners

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

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

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

20. Object-oriented Programming [Electronic resource]. – Acces mode : https://pns.hneu.edu.ua/course/view.php?id=5528