

**Syllabus of the educational discipline**  
**«OBJECT-ORIENTED PROGRAMMING»**

<i>Cycle of Higher Education</i>	<i>First cycle of higher education (Bachelor's degree)</i>
<i>Field of Study</i>	<i>12 Information Technologies</i>
<i>Specialty</i>	<i>123 Computer engineering</i>
<i>Educational program</i>	<i>Computer systems and networks</i>
<i>Discipline status</i>	<i>Normative</i>
<i>Teaching language</i>	<i>English</i>
<i>Year of studies, semester</i>	<i>2 year (3 semester)</i>
<i>Number of credits ECTS</i>	<i>3 credits</i>
<i>Distribution by types of trainings and hours of study</i>	<i>Lectures, Laboratory studies, Independent training</i>
<i>Form of final assessment</i>	<i>Exam</i>
<i>Teacher</i>	<i>Mulesa Oksana Yuriivna, associate professor of department of cybernetics and applied mathematic</i>
<i>Teacher's contacts</i>	<i>oksana.mulesa@uzhnu.edu.ua</i>
<i>Course Schedule</i>	<i>According to the timetable</i>
<p>The purpose of the course is to give students the fundamental basic training necessary to solve both theoretical and practical problems using object-oriented programming languages; to help students in the formation of personality, his intelligence and ability to logical and algorithmic thinking; to educate students in computer culture, understanding the role and place of computer science in modern civilization and world culture; to develop the ability to extend computer knowledge and apply it to analysis and solve problems independently.</p> <p>As a result of studying the discipline the student must:</p> <p><i>know:</i></p> <ul style="list-style-type: none"> <li>- search for the necessary information according to the received task of laboratory work and additional tasks, analyze and evaluate it;</li> <li>- design the architecture of the selected software application, the structure of classes, modules, components, a clear understanding of the distribution of functions of different architectural levels of the application, choose design and implementation methods that are most appropriate to achieve the objectives. Know the basics of structural, modular and object-oriented programming in C++</li> </ul> <p><i>be able to:</i></p> <ul style="list-style-type: none"> <li>- create programs to solve a wide variety of engineering practice problems. Possess modern programming languages, know the latest programming techniques, and learn how to create Windows applications;</li> <li>- present, discuss, and defend their own choices and ways of implementing projects completed orally or electronically as part of laboratory work and individual tasks and with the help of information and communication technology</li> </ul>	
<p><b>Prerequisites for learning</b>            Programming, Data structures and algorithms</p>	
<p><b>Content of the educational discipline</b></p>	
<p><b>Module 1.</b>  <b>Content module 1.</b> An introduction to object-oriented programming.  <b>Topic 1.</b> History of OOP development  <b>Topic 2.</b> OOP  <b>Content module 2.</b> Fundamentals of object-oriented programming.  <b>Topic 3.</b> Concept of classes and objects  <b>Topic 4.</b> Class definition. Working with constructors and destructors  <b>Topic 5.</b> Static methods and fields  <b>Content module 3.</b> Introduction to Polymorphism  <b>Topic 6.</b> Operations overload  <b>Topic 7.</b> Use of friendly structures</p>	

**Content module 4. Inheritance**  
**Topic 8.** Types of inheritance. Deriving classes  
**Module 2.**  
**Content module 5.** Work with abstract classes  
**Topic 9.** Virtual methods  
**Topic 10.** Use of abstract classes  
**Content module 6.** Exception handling  
**Topic 11.** Basic concepts of exception handling  
**Content module 7.** Class templates  
**Topic 12.** Creating and using class templates

<b>Course page on the Moodle platform (personal training system)</b>	<i>Syllabus of the educational discipline, hyperlinks to electronic publications of the discipline, recommended literature, students' attendance, lecture materials, presentations, questions for self-control, methodical materials for laboratory works, tests, tasks for checking students' knowledge.</i> <a href="https://moodle.uzhnu.edu.ua">https://moodle.uzhnu.edu.ua</a>
--	---

**Recommended literature**

1. Robert Lafore, *Object Oriented Programming in C++*. - Sams; Subsequent edition, 2001, - 1012p
2. Bjarne Stroustrup, *The C++ Programming Language, 4th Edition*, Addison-Wesley Professional, 2013. - 1376p
3. Microsoft Overview of Windows Programming in C++ URL: <https://docs.microsoft.com/en-us/cpp/windows>
4. Working Draft, *Standard for Programming. Language C++. ISO/IEC N4582, 1514 pp.* URL: <https://www.open-std.org/jtcl/sc22/wg21/docs/papers/2016/n4582.pdf>

**Assessment system of learning outcomes**

*The ECTS grade that a student receives after studying a credit module of a discipline is determined according to the student's rating. A student's credit module rating consists of the points the student receives during the semester for the following types of work:*

1. Modular control work (MCW) duration of 2 acad. hours each. The maximum number of points for the MCW is 40 points.
2. Performance of laboratory works.

*During an academic semester students perform 6 laboratory works, which are rated a maximum of 50 points.*

*Scores on individual and independent work of students are awarded for: preparation of essays, modernization of tasks, creative approach to task performance, performance of tasks to improve didactic materials on the discipline: 0-10 points for each module.*

*Each module is assessed a maximum of 100 points. At the end of the discipline a rating score is derived as the arithmetic average of the points from the two modules.*

**ECTS and national grading scale**

Mark scale	ECTS	Exam	Test
90 - 100	A	Excellent	Satisfied
82 - 89	B	Good	
74 - 81	C		
64 - 73	D	Satisfactory	
60 - 63	E		
35 - 59	FX	“Unsatisfactory” with possibility to pass the exam again	“Not satisfied” with possibility to pass the exam again
1 - 34	F	“Unsatisfactory” with obligatory repeated study of the discipline	“Not satisfied” with obligatory repeated study of the discipline