

THE BASICS OF OBJECT-ORIENTED PROGRAMMING

(course title)

**COURSE SYLLABUS ABSTRACT**

**of higher education institution**

**speciality**

1-40 05 01 - Information systems and technology (by area)

(speciality code and name)

1-40 05 01 - Information systems and technology (in design and production)

(specialisation code and name)

	STUDY MODE	
	full-time	part-time
Year	<b>2</b>	<b>2</b>
Semester	3, 4	3
Lectures, hours	68	6
Practical classes (seminars), hours		
Laboratory classes, hours	68	6
In-class test (semester, hours)		
Course paper, semester		
Course project, semester	4	3
Pass/fail, semester		
Exam, semester	3, 4	3
Contact hours	136	12
Independent study, hours	172	296
Total course duration in hours / credit units	308/8	

1. Course outline

The aim of Object Oriented Programming is to teach students how to build complex programmes and systems using object-oriented programming.

2. Course learning outcomes

Upon completion of the course, students will be expected to know:

- basic language concepts and syntax, OOP technology and programme development techniques;
- methods for defining and using basic objects and language constructions;
- technology of organization and use of hierarchy of classes, predefined classes and data types, methods of access limitation and handling of exceptional situations;
- methods of classes parameterization and their use to solve problems;
- methods of templates and container abstractions application;
- threading and development of multithreaded applications;

be able to:

- define abstractions, modules, build a hierarchy of classes to implement programmes;
- use methods: typing, encapsulation, inheritance, polymorphism for software development
- use the capabilities of standard libraries;
- use the exception mechanism to create stable applications;
- create their own and use provided standard libraries of templates of complex data structures;
- Use OOP technology to develop complex programs and systems;

possess:

- methods and tools and systems for developing object-oriented software;
- techniques for creating object-oriented software components and organising their interaction in software projects.

3. Competencies

AC-1 Be able to apply basic scientific-theoretical knowledge to solve theoretical and practical problems.

AC-2 Be able to apply systems and comparative analysis.

AC-3 Have research skills.

AC-4 Be able to work independently.

AC-5 Be able to generate new ideas (have creativity).

AC-6 Have an interdisciplinary approach to problem solving.

AC-7 Have skills related to the use of technical devices, information management and computer work.

AC-8 Have oral and written communication skills.

AC-9 Be able to learn and develop their skills throughout their lives.

AC-10 Use the basic laws of science in professional activities.

AC-11 Master the main methods and techniques of acquisition, storage and processing of information by means of computer technology.

AC-14 Organise their own work on a scientific basis and evaluate the results of their activity independently.

SLC-6 Be able to work in a team

PC-1 Master the modern methods, languages, technologies and tools for design and development of software products.

PC-3 Analyse and justify the choice of technical, software tools and systems for the automated support of processes of professional activity.

PC-4 Develop software tools and systems to provide automated support for the solution of problems of professional activity.

PC-6 Carry out testing of software products and applied software tools for compliance with technical requirements.

PC-21 Analyse and evaluate data collected

PC-24 Use global information resources

4. Requirements and forms of midcourse evaluation and summative assessment

The module-rating system for assessing students' knowledge is used in the study of the discipline. The following forms of classes are used: traditional lectures and multimedia lectures, problem / problem-oriented laboratory classes with the use of a computer. The results of the laboratory work are defended.