Syllabus
Course Program

Object-Oriented Programming

Specialty
122 – Computer Science

Institute
Institute of Computer Science and Information Technology

Educational program
Computer Science and Intelligent Systems

Department
Software Engineering and Management Intelligent Technologies (321)

Level of education
Bachelor's level

Course type
Special (professional), Mandatory

Semester
3

Language of instruction
English, Ukrainian

Lecturers and course developers

Olena Nikulina

olena.nikulina@khpi.edu.ua
Doctor of Technical Sciences, Professor, Professor of the Department of Software Engineering and Information Technologies of NTU "KhPI"

Prepared and published more than 100 scientific and educational works (Google Scholar: https://scholar.google.com/citations?user=ZEe2GlcAAAAJ; ORCID https://orcid.org/0000-0003-2938-4215; Scopus: https://www.scopus.com/authid/detail.uri?authorId=57541344600).
More about the lecturer on the department's website

General information

Summary

Acquaintance of students with the basics of software design; object-oriented programming technologies; techniques of working with visual programming environments; acquisition of skills in developing and testing software products running under modern operating systems; formation of abstract thinking in students, which should help solve applied problems related to various fields of knowledge. This course will allow you to understand the current state and latest trends in object-oriented programming. You will learn what classes and objects are, how to handle operation overloading, exception handling, writing interfaces and abstract classes, and using containers and patterns.

Course objectives and goals

Acquisition of the necessary knowledge to master modern technologies of object-oriented analysis, design and programming of an object-oriented model in various programming languages.

Format of classes

Lectures, laboratory work, coursework, independent work, consultations. The final control is an exam.

Competencies

GC1. Ability to think abstractly, analyze and synthesize.
GC2. Ability to apply knowledge in practical situations.

GC3. Knowledge and understanding of the subject area and understanding of professional activities.
GC6. Ability to learn and master modern knowledge.
GC9. Ability to work in a team.
GC8. Ability to design and develop software using various programming paradigms: generalized, object-oriented, functional, logical, with appropriate models, methods and algorithms of computation, data structures and control mechanisms.

Learning outcomes

PLO5. Design, develop and analyze algorithms for solving computational and logical problems, evaluate the effectiveness and complexity of algorithms based on the use of formal models of algorithms and computable functions.
PLO9. Develop software models of subject environments, choose a programming paradigm from the standpoint of convenience and quality of application for the implementation of methods and algorithms for solving problems in the field of computer science.
PLO14. Apply knowledge of methodology and CASE tools for designing complex systems, methods of structural analysis of systems, object-oriented design methodology in the development and study of functional models of organizational, economic, production and technical systems.

Student workload

The total volume of the course is 120 hours (4 ECTS credits): lectures - 32 hours, laboratory classes - 32 hours, self-study - 56 hours.

Course prerequisites

Algorithmization and programming
Algorithms and data structures

Features of the course, teaching and learning methods, and technologies

Teaching and learning methods:
interactive lectures with presentations, discussions, laboratory classes, teamwork, case method, student feedback, problem-based learning.
Forms of assessment:
written individual assignments for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), express surveys (CAS), online tests (CAS), final/semester control in the form of a semester exam, according to the schedule of the educational process (FAS).

Program of the course

Topics of the lectures

Topic 1: Basics of building C++ objects
Creating and using C++ classes.
Encapsulation.
Overloading of operations.
Composition of classes.
Topic 2. Inheritance
Open and closed inheritance.
Handling exceptions
Topic 3. Polymorphism
Virtual and abstract classes
Templates.
Topic 4. Standard classes
Using the tools of the C++ standard library.
C++ containers.
C++ algorithms.
Topic 5. Fundamentals of programming in C#

Basic syntax of the C# language.
Working with arrays and strings in Java.
Creating classes in C#
Topic 6: Encapsulation and inheritance in C#
Nested classes.
Composition.
Inheritance.
Exceptions.
Interfaces and abstract classes.
Using polymorphism.
Generalization.
Working with generics and collections in C#.
Topic 7. Functional and declarative parallel programming
Delegates.
Events.
Lambda expressions.
Graphical user interface.
Graphical tools of .NET.
Declarative programming.
LINQ technology.
Windows Presentation Foundation technology.
Topic 8: UML diagrams and patterns
Using the Unified Modeling Language (UML)
Basics of describing and using design patterns.
Determination of the structure, mechanisms for creating and interacting with classes that allow you to apply typical solutions defined by design patterns.

Topics of the workshops

Workshops are not provided within the discipline.

Topics of the laboratory classes

Topic 1: Working with classes in C++.
Topic 2. Working with the base class and with the descendant class
Topic 3: Abstracts and virtual classes.
Topic 4. Working with classes in C#
Topic 5. Creating class hierarchies.
Topic 6: Using container classes.
Topic 7. Developing a graphical user interface application.
Topic 8. UML diagrams.

Self-study

The plan includes a coursework.
During the course work, it is necessary to design and implement a graphical user interface program that allows you to solve a specific problem. It is necessary to implement data entry from a file, editing and saving data in another file, as well as generating a report on the results of the program.
The program should be implemented using object-oriented technologies. The use of polymorphism is mandatory. To increase the reliability of the program, you must use an exception handling mechanism.
The topic of the course work: development of a graphical user interface application program for processing a database with various information.
The evaluation is based on the following criteria:
1) understanding, degree of mastery of the theory and methodology of the problems under consideration;
2) the degree of mastery of the work material;
3) implementation of a software product on the topic of the course work;
4) testing and demonstration of a graphical user interface program that allows you to solve a specific data processing task;

5) logic, structure, style of presentation of material in written works and in classroom presentations, ability to justify one's position, generalize information and draw conclusions.
The grade of "excellent" is assigned if the student's completed assignment or oral response meets all five of these criteria.
The absence of a particular component reduces the grade by the corresponding number of points.
The assessment pays attention to the quality and independence of the student's work, as well as the timeliness of submitting the completed assignments to the teacher (according to the schedule of the educational process). If any of the requirements are not met, the grade will be lowered.

Course materials and recommended reading

Key literature

1. Ivanov E.O., Linder Y.M., Zhereb K.A. Fundamentals of the C++ programming language: a textbook. - K.: Logos, 2020. - 90 p.
2. Stanley B. Lippman, Josee Lajoie C++ Primer. Fifth Edition. - Addison-Wesley, 2018.
3. Burlakov A. A. Object-oriented analysis and design. Methodical recommendations for independent study of the discipline by students majoring in Software Engineering / A. A. Burlakov: KhNU, 2017. 136 p.
4. Troelsen A. Japikse P. Pro C# 9 with .NET 5: Foundational Principles and Practices in Programming: 10th edition, Apress, 2021, 1411 p.
5. Konovalenko I.V. Programming in C# 7.0: a textbook / Konovalenko I.V., Marushchak P.O., Savkiv V.B. - Ternopil: Ternopil National Technical University named after Ivan Pului, 2017 - 300 p.
6. Price M. J. C# 9 and .NET 5 - Modern Cross-Platform Development: Build intelligent apps, websites, and services with Blazor, ASP.NET Core, and Entity Framework Core using Visual Studio Code: 5th Edition, Packt Publishing, 2020, 822 p.

Additional literature

7. Nikulina O.M. Fundamentals of programming in a visual environment. Methodical instructions for laboratory classes in the course "System Programming" / O. M. Nikulina - Kharkiv: NTU "KhPI", 2014. - 56 p.
8. Nikulina O. M., Kotsiuba N. V. Object-oriented programming in C ++: methodical instructions for laboratory classes in the course "Object-oriented programming" for students majoring in 122 - Computer Science, 126 - Information Systems and Technologies. Kh. : NTU "KhPI", 2022. - 68 p.
9. Nikulina O.M., Ivanov L.V., Kotsiuba N.V. Object-oriented programming in C#: Methodical instructions for laboratory classes in the course "Object-Oriented Programming" for students majoring in 122 - Computer Science, 126 - Information Systems and Technologies. Kh. : NTU "KhPI", 2022. - 64 p.

Assessment and grading

Criteria for assessment of student performance, and the final score structure

The final grade in the discipline is calculated as an average of several components, taking into account the grades of each type of control (grades for laboratory work, grades for coursework and exam grades).
100% final assessment in the form of an exam (18%) and a current assessment (82%).
18% exam
82% current assessment:
Coursework (10%)
Laboratory work (72%)
Laboratory work №1 (9%)
Laboratory work №2 (9%)
Laboratory work №3 (9%)
Laboratory work №4 (9%)
Laboratory work №5 (9%)
Laboratory work №6 (9%)
Laboratory work №7 (9%)
Laboratory work №8 (9%)

Grading scale

Total points	National	ECTS
90–100	Excellent	A
82–89	Good	B
75–81	Good	C
64–74	Satisfactory	D
60–63	Satisfactory	E
35–59	Unsatisfactory (requires additional learning)	FX
1–34	Unsatisfactory (requires repetition of the course)	F

Norms of academic integrity and course policy

The student must adhere to the Code of Ethics of Academic Relations and Integrity of NTU "KhPI": to demonstrate discipline, good manners, kindness, honesty, and responsibility. Conflict situations should be openly discussed in academic groups with a lecturer, and if it is impossible to resolve the conflict, they should be brought to the attention of the Institute's management.
Regulatory and legal documents related to the implementation of the principles of academic integrity at NTU "KhPI" are available on the website: http://blogs.kpi.kharkov.ua/v2/nv/akademichna-dobrochesnist/

Approval

Approved by	08.06.2023	Head of the department Ihor HAMAIUN
	08.06.2023	Guarantor of the educational program Andrii KOPP