

O.M. BEKETOV NATIONAL UNIVERSITY OF URBAN ECONOMY IN KHARKIV

Educational and Scientific Institute of Energy, Information and Transport Infrastructure

APPROVED

Director



(Ihor BILETSKYI)

2021

WORK PROGRAM OF THE ACADEMIC DISCIPLINE

Programming


type of discipline, code for EP	<i>compulsory, PP09</i>
semester	<i>1, 2</i>
number of ECTS credits	<i>9 (5 + 4)</i>
form of final control	<i>differentiated test / exam</i>
language of instruction, teaching and assessment	<i>English</i>
department	<i>Computer Science and Information Technology</i>

for higher education applicants:

level of higher education	<i>first (bachelor's degree)</i>
branch of knowledge	<i>12 Information Technology</i>
specialty	<i>122 Computer Science</i>
educational program	<i>Computer Science</i>
form of study	<i>full-time</i>

2021 – 2022 ACADEMIC YEAR

Developers:

Surname and initials	Position, email	Scientific degree, academic title	Signature
Vladyslav PLIUHIN	Professor, vladyslav.pliuhin@kname.edu.ua	Dr.Sc., Full Professor	

The work programme was approved **at the proceedings** of the Department *Computer Science and Information Technology* _____

Minutes dated « 30 » 08 2021 No. 2

Head of the Department _____ (Maryna NOVOZHYLOVA)

The work program of the discipline corresponds to the Educational Program
Computer Science

Guarantor of the Educational Programme _____ (Mykola PAN)

1. The purpose of the discipline

The purpose of teaching the discipline "Programming" is the theoretical and practical training of students in the field of software development using object-oriented model of the C ++ programming language and the Visual Studio platform for developing Windows applications. The knowledge gained as a result of mastering the discipline will help in the development of system software components of modern information and calculation programs, in the design and implementation of system components of operating systems to such an extent that students can choose implementation tools, find necessary software and technological solutions for practical systems. and subject-oriented tasks.

2. Interdisciplinary connections

The study of this discipline is directly based on:
Higher mathematics, web design, operating systems, computer systems architecture, systems analysis, computer networks, databases.

3. Learning outcomes

Program learning outcome	Teaching methods	Forms of evaluation
LO5. To design, develop and analyze algorithms for solving computational and logical problems, to evaluate the efficiency and complexity of algorithms based on the use of forward models of algorithms and calculated functions.	<ul style="list-style-type: none"> • • • • • notes • work 	<ul style="list-style-type: none"> • Verbal • Visual • Practical • Lecture • Individual • interview • tasks (C++ program) • virtual environment CISCO • Calculation and Graphic Task (CGT) • test •

Program learning outcome	Teaching methods	Forms of evaluation
<p>LO9. To develop software models of subject environments, to choose the programming paradigm from the stand-in of convenience and quality of application for the implementation of methods and algorithms for solving problems in the field of computer science.</p>		

4. Discipline program

Module 1. Introduction to programming

Content module 1.1. The concept of algorithm and typical algorithmic programming structures

The task of this content module is to master the basic skills of writing procedural programs in C ++. Information on the program structure, functions, operators, types of variables is provided. Questions about approaches to the organization of cycles of different types, branches and use of logical operations are raised. In the practical classes, the interface of the Microsoft Visual Studio software development environment and the creation of a simple MFC application using the basic controls of dialog boxes are studied.

Content module 1.2. Organization of programs

The content module provides information on defining functions, passing arguments and returning values, recursion. The work with data transfer by value, address and link is explained. Overloaded functions, built-in functions, global changes, and constant arguments are considered. Arrays, structures, enumerations and data collections are studied. Practical classes are devoted to working with advanced controls for dialog boxes in Microsoft Visual Studio.

Content module 1.3. Basic components of high-level programming languages

The content module reveals the basic principles of object-oriented approach to writing programs in C ++. The concepts of objects and classes, encapsulation, inheritance, operation overload, and type conversion are covered. Issues of polymorphism, abstract and friendly classes, as well as interfaces are revealed. The practical classes cover the creation of a toolbar and timelines, as well as the architecture of the document / presentation in Microsoft Visual Studio.

Module 2. Software design technology

Content module 2.1. Data structures

The content module contains detailed information about addresses and pointers. Address retrieval operations, pointers to arrays, functions, strings, objects, and other pointers are covered. Information about virtual functions and their components is given. The practical classes focus on working with text and graphics in Microsoft Visual Studio.

Content module 2.2. Working with files

The content module includes questions about streaming classes, streaming input / output of disk files, error handling. Provides information on working with HTTP / HTTPS protocols and handling hyperlinks. The basic principles of working with databases and SQL queries are considered. The practical classes provide theoretical information of lectures on the implementation of file operations I / O, work with HTTP / HTTPS protocols, import and export of data and serialization in Microsoft Visual Studio, as well as work with cloud storage. The work with the Microsoft Access database and the ODBC service using SQL queries is considered.

Content module 2.3. Algorithmization of typical computational problems

The content module covers the main elements (sequential and associative containers, functional objects, etc.) from the standard library of C ++ templates. Provides information about directives and preprocessor operators, as well as the use of the namespace. Practical classes are devoted to the study of all stages of development of a complex application in the environment of Microsoft Visual Studio with the development of a help system in HTML.

5. Structure of the discipline and the distribution of time

Content modules	Number of hours				
	Total	Lect.	Pract.	Lab.	Ind. work
MODULE 1 (autumn semester)	150	30	30	-	90
Content module 1.1	35	10	10	-	15
Content module 1.2	35	10	10	-	15
Content module 1.3	35	10	10	-	15
Individual Task	30	-	-	-	30
Final control	15	-	-	-	15
MODULE 2 (spring semester)	120	34	34	-	52
Content module 2.1	35	12	12	-	11
Content module 2.2	35	12	12	-	11
Content module 2.3	35	10	10	-	15
Final control	15	-	-	-	15

6. Themes of the lectures

Module 1. Introduction to programming

Theme	Contents (plan)	Number of aud. hours
CM 1.1. The concept of algorithm and typical algorithmic programming structures		
1. Introduction to programming	Features of the C ++ language. The concept of object-oriented programming. Terms and definitions	2
2. Program structure and types of variables	Functions. Operators. Directives. Types of variables. Type conversion. Arithmetic operations.	2
3. Loops	Loops for, while, do. Examples using cycles	2
4. Branching	Construction if... else. Operators switch... break. Logical operations AND, OR, NO. Priorities of operations. Transition operators break, continue, goto.	2
5. Structures and enumerations	Definition of structure. Nested structures. Enumeration.	2
CM 1.2. Organization of programs		
6. Functions. Part 1	Simple functions. Passing arguments to a function. Return values	2
7. Functions. Part 2	Link to arguments. Overload.	2
8. Arrays and strings. Part 1	Definition of arrays. Access to array elements. Multidimensional arrays. Transfer arrays to functions. Array operations. Examples.	2
9. Arrays and strings. Part 2	Arrays of structures. Defining strings. Arrays of strings. String operations. Examples.	2
10. Operators overload	Overload of unary and binary operators. Arithmetic operators. Comparison operators. Conversion between basic data types.	2
CM 1.3. Basic components of high-level programming languages		
11. Classes and objects. Part 1	Definition of a simple class. Using the class. Constructors and object creation. Destructors.	2
12. Classes and objects. Part 2	Objects as function arguments. UML class diagrams. Examples of programs.	2
13. Classes and objects. Part 3	Structures and classes. Static class data.	2
14. UML class diagrams	The concept of UML charts. Construction of diagrams. Events. Information about object-oriented analysis	2
15. Practical use of objects and classes	Examples of creation and options for using object-oriented programs	2

Module 2. Software design technology

Theme	Contents (plan)	Number of aud. hours
CM 2.1. Data structures		
1. Inheritance	Basic and derivative classes. Hierarchy of classes. General and partial imitation. Multiple imitation. Composition.	2
2. Addresses and pointers. Part 1	Address retrieval operation. Access to a variable by pointer. Void pointers. Pointers and arrays. Pointers-constants and pointers-variables. Row pointers. Memory management	2
3. Addresses and pointers. Part 2	Pointers and arrays. Pointers and functions. Pointers to objects. Related list. Pointers to pointers. UML charts.	2
4. Virtual functions.	Definition of virtual functions. Access to virtual methods through pointers. Late connection. Abstract classes and pure virtual functions. Virtual base classes.	2
5. Friendly and static functions	Definition of friendly functions. Friendly classes. Static functions. This pointer. Dynamic typing.	2
6. Using objects and classes	Examples of the use of imitation. Examples of the use of virtual functions and abstract classes. Examples of using friendly features.	2
CM 2.2. Working with files		
7. Stream classes	Hierarchy of flow classes. Streaming objects. Flow errors. Stream input / output of disk files.	2
8. File operations	File pointers. Error handling. File input / input using methods. Operator overload. CFile class.	2
9. Multi-file programs	Using multi-file programs. Inter-file interaction. Big Data Classes. Asynchronous and parallel flows. High performance (HPC) calculations	2
10. Working with the HTTP / HTTPS protocol	Hyperlink processing. Defining HTTP / HTTPS streams. Query generation. Working with Google Drive	2
11. Working with databases	Working with the ODBC service. Access to Google Drive cloud storage. Formation of SQL-queries.	2
12. Templates and exceptions	Function templates. Class templates. Exclusion.	2
CM 2.3. Algorithmization of typical computational problems		
13. Introduction to STL	Introduction: containers, algorithms and iterators. Algorithms.	2
14. Serial containers	Vectors. Lists. Iterators. Specialized iterators.	2
15. Associative containers	Plural. Multi-sets. Display. Associative array.	2
16. Saving user objects	Sets and list of objects. Create your own functional objects. Behavior of containers.	2
17. Preprocessor namespace and directives	The concept of namespace. Std namespace. Preprocessor directives and operators. Macro names.	2

7. Themes of the practical classes

Module 1. Introduction to programming

Theme	Contents (plan)	Number of aud. hours
CM 1.1. The concept of algorithm and typical algorithmic programming structures		
CISCO Chapter 0	Sandbox and Lab Tool	2
CISCO Chapter 0	Introduction to IDE and online tools	2
CISCO Chapter 1	Labs 1 – 3	2
CISCO Chapter 1	Labs 4 – 6	2
CISCO Chapter 1	Labs 7 – 8	2
CM 1.2. Organization of programs		
CISCO Chapter 2	Labs 1 – 3	2
CISCO Chapter 2	Labs 4 – 6	2
CISCO Chapter 2	Labs 7 – 9	2
CISCO Chapter 2	Lab 10	2
CISCO Chapter 2	Labs 11	2
CM 1.3. Basic components of high-level programming languages		
CISCO Chapter 3	Labs 1 – 3	2
CISCO Chapter 3	Labs 4 – 6	2
CISCO Chapter 3	Labs 7 – 8	2
CISCO Chapter 3	Lab 9	2
CISCO Chapter 3	Lab 10	2

Module 2. Software design technology

Theme	Contents (plan)	Number of aud. hours
CM 2.1. Data structures		
CISCO Chapter 4	Labs 1 – 2	2
CISCO Chapter 4	Labs 3 – 4	2
CISCO Chapter 4	Labs 5 – 6	2
CISCO Chapter 5	Labs 1 – 2	2
CISCO Chapter 5	Labs 3 – 4	2
CISCO Chapter 5	Lab 5	2
CM 2.2. Working with files		
CISCO Chapter 6	Labs 1 – 3	2
CISCO Chapter 6	Labs 4 – 5	2
CISCO Chapter 6	Labs 6 – 7	2
CISCO Chapter 7	Labs 1 – 2	2
CISCO Chapter 7	Labs 3 – 4	2
CISCO Chapter 7	Lab 5	2
CM 2.3. Algorithmization of typical computational problems		
CISCO Chapter 8	Lab 1	2
CISCO Chapter 8	Lab 2	2
CISCO Chapter 8	Lab 3	2
CISCO Chapter 8	Lab 4	2
CISCO Chapter 8	End-of-course Test	2

8. Individual Task (IT)

Calculation and Graphic Task (CGT) on the topic "Development of a personal organizer". The aim of the CGT is to develop skills in object-oriented programming in C ++ in Microsoft Visual Studio in the course of solving a specific practical task - the development of a window MFC application for a personal organizer. At the same time, students have the right to choose their own layout of the program windows, as well as the implementation of the class structure. The purpose of this approach is to develop the ability to make informed decisions during project work and gain basic experience in developing information systems.

9. Methods of control and the procedure for assessing learning outcomes

- Oral interview
- Individual tasks (C++ program)
- Testing in virtual environment CISCO
- Differentiated test
- Exam

Structure of the discipline and the distribution of points

Content modules	Maximum number of points			
	Total	Pract.	Lab.	Ind. Work
MODULE 1	100	45	-	55
Content module 1.1	20	10	-	10
Content module 1.2	15	5	-	10
Content module 1.3	15	5	-	10
Individual Task	20	-	-	20
Final control	30	-	-	30

Content modules	Maximum number of points			
	Total	Pract.	Lab.	Ind. Work
MODULE 2	100	40	-	60
Content module 2.1	25	15	-	10
Content module 2.2	25	15	-	10
Content module 2.3	20	10	-	10
Final control	30	-	-	30

Types of tasks, means of control and maximum number of points

Module 1. Introduction to programming

CM 1.1. The concept of algorithm and typical algorithmic programming structures	20
CISCO Chapter 0 (CISCO report)	5
CISCO Chapter 1 (CISCO report)	5
Individual Task on theoretical material CM 1.1 (C++ program)	10
CM 1.2. Organization of programs	15
CISCO Chapter 2 (CISCO report)	5
Individual Task on theoretical material CM 1.2 (C++ program)	10
CM 1.3. Basic components of high-level programming languages	15
CISCO Chapter 3 (CISCO report)	5
Individual Task on theoretical material CM 1.3 (C++ program)	10
Individual task (CGT)	20
Program development (Report of Calculation and Graphic Task)	10
Presentation and defense of CGT (Power Point presentation)	10
Final control - diff. test	30
Individual Task on theoretical material of the course (C++ program)	20
Theoretical Task (Oral interview)	10
TOTAL FOR THE MODULE 1	100

Module 2. Software design technology

CM 2.1. Data structures	25
CISCO Chapter 4 (CISCO report)	10
CISCO Chapter 5 (CISCO report)	5
Individual Task on theoretical material CM 2.1 (C++ program)	10
CM 2.2. Working with files	25
CISCO Chapter 6 (CISCO report)	10
CISCO Chapter 7 (CISCO report)	5
Individual Task on theoretical material CM 2.2 (C++ program)	10
CM 2.3. Algorithmization of typical computational problems	20
CISCO Chapter 8 (CISCO report)	10
Individual Task on theoretical material CM 2.3 (C++ program)	10
Final control - exam	30
Individual Task on theoretical material of the course (C++ program)	20
Theoretical Task (Oral interview)	10
TOTAL FOR THE MODULE 2	100

Grading scale

The sum of points for all types of educational activities	Score on a national scale	
	for the exam, diff. test	for test
90-100	excellent	passed
82-89	good	
74-81		
64-73		
60-63	satisfactorily	
35-59	unsatisfactory with the possibility of retaking	failed with the possibility of retaking
0-34	unsatisfactorily with mandatory re-study of the discipline	failed with mandatory re-study of the discipline

10. Material and technical and information support

Methodical support

1. Distance course on the Moodle platform 3.8 "Object-oriented programming. Module 1» <https://dl.kname.edu.ua/course/view.php?id=578>
2. Distance course on the Moodle platform 3.8 "Object-oriented programming. Module 2» <https://dl.kname.edu.ua/course/view.php?id=579>
3. CISCO network academy course “CPA: Programming Essentials in C++”. <https://www.netacad.com/courses/programming/essentials-programming-c-plus-plus>

Recommended literature and information resources

1. V. Pliuhin. Programming. Lectures for the foreign students of the 1st year of education on the specialty “Computer Sciences”. – O.M. Beketov NUUEKh. – 2020. – 130 p.
2. R. Lafore. Object-Oriented Programming in C++. SAMS Publishing, 2002, 1038p.
3. Stroustrup, Bjarne. The C++ Programming Language (Fourth ed.). Addison-Wesley. 2013. – 1366 p.
4. Meyers, Scott. Effective C++ (Third ed.). O'Reilly Media, Inc. – 2014. – 334 p.

Hardware, equipment, software products

1. Microsoft Visual Studio 2019 Community Edition.
2. Computer rooms (aud. 218a_cb, 218б_cb, 218B_cb, 226cb).