Syllabus
Course Program

Fundamentals of Programming

Specialty
121 – Software Engineering

Institute
Institute of Computer Science and Information Technology

Educational program
Software Engineering

Department
Software Engineering and Management Intelligent Technologies (321)

Level of education
Bachelor's level

Course type
Special (professional), Mandatory

Semester
1-2

Language of instruction
English, Ukrainian

Lecturers and course developers

Lev Ivanov

lev.ivanov@khpi.edu.ua
Senior Lecturer of Software Engineering and Management Intelligent Technologies Department

Google Scholar: https://scholar.google.com/citations?user=B8fggLEAAAAJ
More about the lecturer on the department's website

Pavlo Smolin

pavel.smolin@khpi.edu.ua
Senior Lecturer of Department of Software Engineering Management Intelligent technologies, National Technical University "Kharkiv Polytechnic Institute

Google Scholar: https://scholar.google.com/citations?user=zCHB-xoAAAAJ&hl%20google%20scholar
ORCID: https://orcid.org/0000-0002-1290-9698
Scopus: https://scholar.google.com/citations?user=zCHB-xoAAAAJ&hl%20google%20scholar
More about the lecturer on the department's website

General information

Summary

The task of the discipline is to acquire the necessary level of knowledge about the principles of algorithmization, the basic syntax of the C++ programming language, procedural and modular approaches, callback mechanisms, the object-oriented model of the C++ language, the application of the object-oriented approach, as well as methods of generalized programming.

Course objectives and goals

Obtaining the necessary knowledge on mastering the theoretical foundations of the C++ programming language and acquiring practical skills in its use during the development of programs based on the principles of structural, procedural-oriented, object-oriented and generalized programming.

Format of classes

Lectures, laboratory classes, self-study, consultations. Final control in the form of an exam.

Competencies

К01. Ability to think abstractly, analyze and synthesize.
К02. Ability to apply knowledge in practical situations.
К05. Ability to learn and master modern knowledge.
К06. Ability to search, process and analyze information from various sources.
К14. Ability to participate in software design, including modeling (formal description) of its structure, behavior and processes of functioning.
К25. Ability to reasonably choose and master the tools for software development and maintenance.
К26. Ability to think algorithmically and logically.

Learning outcomes

PLO01. Analyze, purposefully search and select information and reference resources and knowledge necessary for solving professional problems, taking into account modern achievements of science and technology.
PLO07. To know and apply in practice the fundamental concepts, paradigms and basic principles of functioning of language, tools and computing tools of software engineering.
PLO08. Be able to develop a human-machine interface.
PLO15. Motivated to choose programming languages and development technologies to solve the problems of creating and maintaining software.
PLO23. Be able to document and present the results of software development.

Student workload

The total volume of the course is 330 hours (11 ECTS credits): lectures – 90 hours, laboratory classes – 90 hours, self-study – 150 hours.

Course prerequisites

Fundamentals of software engineering

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 Computer science. Software. Development of algorithms

Basic concepts of computer science. Numerical systems. Software. Operating Systems. Text and binary files. Basic concepts of programming. Algorithms. Graphic presentation of algorithms. Classification of algorithms.

Topic 2 Basic syntax of C++. Data types and operations

The main characteristics of the C++ programming language. Stages of program development in C++ language. The first program is in C++. Basic elements of the C++ programming language. Fundamental data types. Constant values. Definition of variables and named constants. Expressions and operations. Mathematical operations. Compound assignment, increment and decrement. Relational operations. Logical operations. Conditional operation. Using bitwise operations. Using operations for data input and output

Topic 3 C++ statements. Branching. Cycles

An empty statement. An expression statement. Compound statement. Branching statements. Cyclic statements. Transition statement

Topic 4 Creating and calling functions

Function declaration and definition. The void returning type. Scope. Static local variables. Recursion. Inline functions. Function name overloading. Default arguments. References

Topic 5 Description and use of arrays

Definition of one-dimensional arrays. Range type. Use of one-dimensional arrays. Using a loop built on a range. Multidimensional arrays. Arrays as function parameters

Topic 6 Pointers, strings and files

Definition of pointers. Arrays and pointers. Using free store. Character arrays. Using C-style input and output. Working with files

Topic 7 Callback. Physical and logical structure of the program

Typedefs. Pointers to functions. Callback. Using header files. Include Guards. Namespaces

Topic 8 Creating and using custom types

Custom types. Enumerations. Structures. Unions. Sorting arrays of structures using pointers. Using linked lists

Topic 9 Creating classes. Class members

Prerequisites for the emergence of an object-oriented approach. Definition of classes. Constructors and destructors. Class scope. Static class members. Class friends. Operator overloading. Exception handling. Class composition

Topic 10 Inheritance and polymorphism. Templates

Using UML to represent classes. Inheritance. Features of multiple inheritance. Exception class hierarchies. Polymorphism. Virtual functions. Abstract classes. Using templates. Template functions. Class templates

Topic 11 Using the C++ Standard Library, lambda expressions and modules

The general structure of the C++ Standard Library. Standard sequential containers. Iterators. Strings. Sequence adapters. Associative arrays and sets. Algorithms of the Standard Library. Functional objects. Lambda expressions. Using modules

Topics of the workshops

Workshops are not provided within the discipline.

Topics of the laboratory classes

Topic 1 Design of algorithms
Topic 2 Data types and operators
Topic 3 C++ statements
Topic 4 Use of functions
Topic 5 Use of arrays and pointers
Topic 6 Working with pointers, strings, and files
Topic 7 Pointers to functions and header files
Topic 8 Working with enumerations and structures
Topic 9 Creation and use of C++ classes
Topic 10 Using polymorphism and templates
Topic 11 Use of the standard template library

Self-study

The curriculum includes the completion of coursework (CW). At the beginning of the semester, students choose the topics of the course work from the list or propose their own topics and agree them with the teacher. The CW is completed during the semester and is defended during the test week or examination session. Students are recommended with additional materials for self-study and processing.

Course materials and recommended reading

Key literature

1. Stroustrup B. The C++ Programming Language: 4th Edition, Addison-Wesley, 2013, 1368 p.
2. Lippman S. B., Lajoie J., Moo B. E. C++ Primer: 6th Edition, Addison-Wesley Professional, 2011. 992 p.
3. Schildt H. C++: The Complete Reference: 4th Edition, McGraw-Hill Education, 2002, 1056 p.
4. Трофименко О.Г. С++. Алгоритмізація та програмування : підручник / О.Г. Трофименко, Ю.В. Прокоп, Н.І. Логінова, О.В. Задерейко. 2-ге вид. перероб. і доповн. – Одеса : Фенікс, 2019. – 477 с.
5. Пекарський Б.Г. Основи програмування: Навчальний посібник. Кондор, 2018. - 364 с.

Additional literature

1. Eckel B. Thinking in C++, Vol. 1: Introduction to Standard C++: 2nd Edition, Prentice Hall, 2000, 840 p.
2. Воловщиков В.Ю., Іванов Л.В., Рубін Е.Ю., Гончаренко Т.Г.. Мова С++ в програмуванні та комп'ютерних науках. – Харків: ФОП Мезіна В.В., 2017. – 280 с.

Assessment and grading

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

100% of the final grade consists of the results of the assessment in the form of an exam (40%) and current assessment (60%):
- 11 laboratory works (3% each);
- coursework (27%).

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 11.04.2023 Head of the department
Ihor HAMAIUN

Guarantor of the educational program
Uliya LITVINOVA