Syllabus
Course Program

Fundamentals of Programming

Specialty	**Institute**
121 – Software Engineering	Institute of Computer Science and Information Technology
Educational program	**Department**
Software Engineering	Software Engineering and Management Intelligent Technologies (321)
Level of education	**Course type**
Bachelor's level	Special (professional), Mandatory
Semester	**Language of instruction**
1-2	English, Ukrainian

Lecturers and course developers
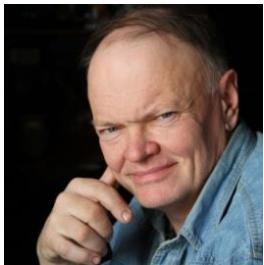
Lev Ivanov

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

General information

Summary

The objective of the discipline is to acquire the knowledge and skills necessary for the design and development of relational databases using client-server database management systems (DBMS) on the example of MySQL, implementation of business logic tools in a database based on stored procedures, functions and triggers, ensuring data integrity, maintaining data consistency and security based on transaction mechanisms and user rights, as well as building client-server applications for working with databases using PHP (mysqli, PDO), Java (JDBC), and Python (PyMySQL, MySQL Connector) as examples.

Course objectives and goals

Developing students' theoretical and practical knowledge necessary for the design and development of databases (DB) in solving problems related to the development, maintenance, and quality assurance of software.

Format of classes

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

Competencies

K01. Ability to think abstractly, analyze and synthesize.
K02. Ability to apply knowledge in practical situations.
K05. Ability to learn and master modern knowledge.
K06. Ability to search, process and analyze information from various sources.

K14. Ability to participate in software design, including modeling (formal description) of its structure, behavior and processes of functioning.
K25. Ability to reasonably choose and master the tools for software development and maintenance.
K26. 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

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
Definition of pointers. Arrays and pointers. Using free store. Character arrays. Using C-style input and output. Working with files
Typedefs. Pointers to functions. Callback. Using header files. Include Guards. Namespaces
Custom types. Enumerations. Structures. Unions. Sorting arrays of structures using pointers. Using linked lists
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
Using UML to represent classes. Inheritance. Features of multiple inheritance. Exception class hierarchies. Polymorphism. Virtual functions. Abstract classes. Using templates. Template functions. Class templates
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

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 C++ 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
	08.06.2023	Guarantor of the educational program Uliya LITVINOVA

National Technical University "Kharkiv Polytechnic Institute"