Syllabus
Course Program

Algorithmization and 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
1-2

Language of instruction
English, Ukrainian

Lecturers and course developers

Mariia Bilova

mariia.bilova@khpi.edu.ua
PhD, Candidate of Engineering Sciences, Associate Professor of Department of Software Engineering and Information Technology Management.

Number of scientific and educational publications more than 50 (Google Scholar https://scholar.google.com/citations?user=b3YLGToAAAAJ; ORCID ID - https://orcid.org/0000-0001-7002-4698; Scopus ID https://www.scopus.com/authid/detail.uri?authorId=57190442390
More about the lecturer on the department's website

Svitlana Kovalenko

Svitlana.Kovalenko@khpi.edu.ua
Candidate of Engineering Sciences, Associate Professor, Associate Professor of Software Engineering and Management Intelligent Technologies Department

Google Scholar: https://scholar.google.com/citations?user=jeD1w74AAAAJ&hl
ORCID: https://orcid.org/0000-0001-6770-6778
Scopus: https://www.scopus.com/authid/detail.uri?authorId=57212035934
Web of Science: https://www.webofscience.com/wos/author/record/F-8252-2017)
More about the lecturer on the department's website

Natalia Fonta

Natalia.Fonta@khpi.edu.ua
PhD, Associate Professor, Associate Professor of SEMIT Department. Number of scientific and educational publications is more than 60.
(Google Scholar:
https://scholar.google.com.tw/citations?hl=ru&pli=1&user=we3S6nwAAAAJ;
Scopus: https://www.scopus.com/authid/detail.uri?authorId=57215861869;
ORCID: https://orcid.org/0000-0001-5593-1409).

Leading lecturer in disciplines: "Probability theory and mathematical statistics"," Numerical Methods ".
Scientific directions: development of information systems for strategic company management; application of computer intelligence methods and models for solving problems of managing complex organizational systems; business analytics.
More about the lecturer on the department's website

General information

Summary

The course covers a broad range of topics related to programming, algorithms, data structures, and computer science in general, while fostering analytical skills, critical thinking, and a creative approach to problem-solving. Students learn the fundamental concepts of programming using the C language and also get acquainted with higher-level languages like Python and JavaScript.
Throughout the course, you will gain practical skills by completing assignments and developing your own projects.
The foundation of the course is based on CS50, a well-known introductory computer science course taught at Harvard University. Students who successfully complete all the assignments will have the opportunity to receive a certificate of completion endorsed by the course instructor, Professor David Malan.

Course objectives and goals

The course aims to develop students' algorithmic thinking and acquire programming skills in languages such as C and Python to solve practical problems in various subject areas, in accordance with modern principles and trends in software development.

Format of classes

Lectures, laboratory classes, consultations, self-study. Final control in the form of 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.
GC7. Ability to search, process and analyze information from various sources.
GC9. Ability to work in a team.
PC3. Ability to think logically, build logical conclusions, use formal languages and models of algorithmic computing, design, develop and analyze algorithms, evaluate their effectiveness and complexity, solvability and intractability of algorithmic problems for adequate modeling of subject areas and creation of software and information systems.
PC8. 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

PLO1. To apply knowledge of the basic forms and laws of abstract and logical thinking, the basics of the methodology of scientific knowledge, forms and methods of extracting, analyzing, processing and synthesizing information in the subject area of computer science.
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.

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

A basic understanding of mathematical concepts such as logic, algebra, and statistics, as well as basic computer skills, are required.

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

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

Program of the course

Topics of the lectures

Topic 1. Algorithms and their Development Features
Learning objectives and course specifics. Concept of a black box. Algorithms and algorithmic thinking. Pseudocode. Graphical representation of an algorithm. Activity diagram.
Topic 2. Number Systems
Data representation methods. Concept of a number system. Decimal system. Binary system. Octal system. Hexadecimal system. Conversion between systems. ASCII table. Color models. Basics of the SCRATCH programming language.
Topic 3. Data Types and Variables in C
Concept of a data type. Type system of C language. Visual Studio Code. Data types in C. Concept of a variable. Naming conventions for variables in C. Declaration and initialization of variables. Variable scopes. Operators. Assignment operators. Mathematical operators. Logical operators. Comparison operators.
Topic 4. Conditional Statements
Branching in a program. If statement. If-else statement. If-else if-else statement. Switch statement. Ternary operator.
Topic 5. Loops
Concept of a loop. Usage of loops in C. For loop. While loop. Do-while loop.
Topic 6. Arrays
Concept of an array. Declaration and initialization of arrays. Key characteristics of arrays. Indexing in arrays. String as an array of characters. Multidimensional arrays. Working with multidimensional arrays.
Topic 7. Functions
Concept of a function. Declaration and definition of a function. Function prototype. Function call. Formal and actual parameters of a function. Return statement. Main function. Command-line arguments. Concept of recursion. Components of recursion.
Topic 8. Memory Management
Concept of stack and heap. Pointers. Dynamic memory allocation in C. Usage of sizeof. Memory deallocation. Changing the allocated memory size. File concept. Interacting with files in C. Stream concept.
Topic 9. Structures
Concept of a structure. Creating a structure. Usage of structures in C. Data structures. Concept of a singly linked list and doubly linked list. Associative array. Hash tables and hash functions. Resolving collisions. Implementation specifics of data structures in C.

Semester 2
Topic 1. Python Programming Language. Data Types
Type system of Python language. Data types. Working with variables in Python. Operators: mathematical, logical, assignment, membership, comparison.
Topic 2. Conditions and Loops in Python
Specifics of conditional statements in Python. While loop. For loop.
Topic 3. Functions and String Operations in Python
Functions. Function arguments. Strings in Python. Indexing in strings, negative indexing. Working with substring.
Topic 4. Basics of Databases
Data. Databases. Database management systems. SQLite. Relational databases. Basics of database design.
Topic 5. Basics of SQL
Storing data in CSV files. Basic DDL and DML commands.
Topic 6. Basics of UML
Unified Modeling Language. Tools for developing UML diagrams. Use case diagram. Class diagram. Deployment diagram. Sequence diagram.
Topic 7. Basics of Web Development
Hypertext Markup Language. Structure of a web page. Document Object Model. Block and inline tags. Cascading Style Sheets. CSS rules. Types of selectors. CSS properties.
Topic 8. Basics of Backend Development
HTTP protocol. Flask framework. Simple Flask application. Routing. Template visualization. Working with databases in the application.

Topics of the workshops

Workshops are not provided within the discipline.

Topics of the laboratory classes

Semester 1
Topic 0. Scratch
Topic 1. Basic syntax constructs of the C programming language
Topic 2. Arrays and functions
Topic 3. Algorithms
Topic 4. Memory management
Topic 5. Data structures

Semester 2
Topic 1. Basic concepts of the Python programming language
Topic 2. Fundamentals of SQL
Topic 3. Basics of HTML and CSS
Topic 4. Creating an application using Flask

Self-study

Semester 1
During the first semester, students are asked to complete 2 individual homework assignments:
1) development of an activity diagram according to the variant (5 points);
2) development of a program based on the activity diagram from first individual assignment (5 points).
Semester 2
During the second semester, students must complete a course work in accordance with the chosen topic. The course work must be written using HTML, CSS, Python, and the Flask microframework, and also involves the design and development of a database. This work accounts for 30% of the second semester grade, of which 15 points are awarded for research, design, and supporting documentation in the form of a report, and 15 points for the project and its presentation (defense of the course work). Possible topics for the course work are listed below.
Development of software for training user response

Developing software for generating color palettes based on an image
Development of software for training user's memory
Development of software for tracking user's weight
Development of software for testing student's knowledge of the chosen programming language
Development of software for selecting random movies according to the user's mood
Development of software for organizing user's working time (task tracker)
Development of software for recording data on books read
Development of software for tracking user's mood
Development of software for tracking user's expenses
Development of software for tracking user's playing time
Development of software for scheduling visits to doctors
Development of software for waste separation with displaying a map of places where you can hand in garbage
Development of software for developing user's mindfulness
Development of software for storing and tracking user's goals
Development of software for learning the spelling of new words
Development of software for tracking data on the health of a pet
Development of software to display countries visited by a user
Development of software for calculating the calorie content of a person's diet
Development of software for organizing the process of book exchange
Development of a program for recording the volume of water drunk by the user with the display of statistics
Development of software for applying filters to images
Development of software for tracking classes in the gym
Development of software for calculating compatibility by zodiac sign
Development of software for deepening knowledge of verb forms in English
Development of software for tracking user's progress while studying at a driving school
Another topic is possible at the request of the student, agreed with the teacher

Course materials and recommended reading

Compulsory materials

1. Kornienko M.M., Ivanova I.D. Informatics. Fundamentals of algorithmization and programming. – Ranok, 2011. – 48 p.
2 Cormen T. H. Introduction to Algorithms Third Edition / Thomas H. Cormen Charles E. Leiserson Ronald L. Rivest Clifford Stein // The MIT Press, 2009. – 1313 p.
3. Stephens R. Essential Algorithms: A Practical Approach to Computer Algorithms. – John Wiley & Sons, Inc, 2013. – 544 c.
4. C Programming Absolute Beginner's Guide. Third Edition. - Pearson Education, 2014. – 617c.
5. Seacord R. C. Effective C: An Introduction to Professional C Programming. – No Starch Press, 2020. – 272 p.
6. Thomas Mailund Pointers in C Programming. A Modern Approach to Memory Management, Recursive Data Structures, Strings, and Arrays. – Apress,
7. Anquetil R Fundamental Concepts for Web Development: HTML5, CSS3, JavaScript and much more. – Independently published, 2019. – 276 c.
8. Myers M. Smarter way to learn Python. – 2017. – 234 p.
9. Stephenson B. The Python Workbook. – Springer: Texts in Computer Science, 2019. – 218 p.
10. Matthes E. Python Crash Course. – Old Lion Publishing House, 2021. – 600 p.
11. Rudenko V.D., Zhugastrov O.O. Basics of algorithmization and programming in Python. – Ranok, 2019. – 192 p.
12. Allen G. Taylor Author of SQL All-in-One For Dummies. 9th edition. Hoboken, 2019. – 496 p.
13. Upadhyay K. Ch. HTML5 For Web Designers. Complete Hypertext Markup Language Guidance. - Independently published, 2020. – 71 p.
14. Grant K. J. CSS in depth. – Manning Publications Co, 2018. – 445 p.
15. Grinberg M. Flask Web Development: developing web applications with Python. 2nd edition. - O'Reilly Media, Inc, 2018. – 314 p.

Additional materials
16. Skiena S. S. The Algorithm Design Manual. Third edition. – Springer, Texts in Computer Science, 2020. – 810 p.
17. Al Sweigart Invent Your Own Computer Games with Python, 4th edition. – No Starch Press, 2017. – 376 p.
18. Martin R. Clean code. – 2019. – 368 c.
19. Chacon S. Pro Git / Scott Chacon, Ben Straub. – Apress, 2014.– 608 p.

Web-sources
20. The GNU C Reference Manual [Electronic resource]. – Access mode: https://www.gnu.org/software/gnu-c-manual/gnu-c-manual.html
21. Beej's Guide to C Programming [Electronic resource]. – Access mode: https://beej.us/guide/bgc/html//index.html
22. CS50 «Introduction to Computer Science» on Prometheus [Electronic resource]. – Access mode : https://edx.prometheus.org.ua/courses/ Prometheus/CS50/2016_T1/info
23. CS50 [Electronic resource]. – Access mode : https://cs50.harvard.edu/
24. C Programming Language Documentation. [Electronic resource]. – Access mode: https://devdocs.io/c/
25. Front-End Developer Handbook 2018 / Cody Lindley – Frontend Masters. –2018. – 168 p. [Electronic resource]. – Access mode : https://legacy.gitbook.com/book/frontendmasters/front-end-developer-handbook-2018/details.
26. SQL Tutorial [Electronic resource]. – Access mode : https://www.w3schools.com/sql/
27. CSS Snapshot 2017. W3C Working Group Note [Electronic resource]. – Access mode : https://www.w3.org/TR/css-2017/.
28. HTML 5.2. W3C Recommendation [Electronic resource]. – Access mode : https://www.w3.org/TR/html52/.
29. UML [Electronic resource]. – Access mode : https://www.uml.org/
30. Fundamentals of UML [Electronic resource]. – Access mode : https://docs.kde.org/trunk5/uk/umbrello/umbrello/uml

Assessment and grading

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

Semester 1

The final grade consists of grades for six laboratory works with a total of 48 points (Laboratory works 0, 5 - 10 points each, Laboratory works 1-4 - 7 points each), two individual homework assignments with a total of 10 points (5 points each), four tests with a total of 12 points (control works 1-3 - 3 points each, control work 4 - 5 points), and an exam - 30 points. The basis for admission to the exam is the completion of laboratory works 0-4.

Semester 2

The final grade consists of grades for four laboratory works with a total of 30 points (Laboratory works 1-2 - 7 points each, Laboratory works 3-4 - 8 points each), two control works in the form of tests with a total of 10 points (5 points each), a term paper, 30 points, and an exam - 30 points. The basis for admission to the exam is the completion of all laboratory work and course work.

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