



Fundamentals of Programming.

Course work

Syllabus

Requisites of the Course

Cycle of Higher Education	<i>First cycle of higher education (Bachelor's degree)</i>
Field of Study	<i>12 Information Technologies</i>
Speciality	<i>121 Software engineering</i>
Education Program	<i>Software Engineering of Multimedia and Information Retrieval Systems</i>
Type of Course	<i>Normative</i>
Mode of Studies	<i>full-time</i>
Year of studies, semester	<i>1 year (2 semester)</i>
ECTS workload	<i>1 credits</i>
Testing and assessment	<i>2 semester – Test</i>
Course Schedule	
Language of Instruction	<i>English</i>
Course Instructors	PhD, Associate Professor, Yuliia Boiarinova mobile +380671751308, email ub@ua.fm PhD, Associate Professor, Tetiana Zabolotnia tetiana.zabolotnia@gmail.com
Access to the course	https://t.me/+zub7kDn0N2g0Zjc6 https://classroom.google.com

Outline of the Course

1. Course description, goals, objectives, and learning outcomes

*The purpose of studying the discipline "Fundamentals of Programming. Course work" is: consolidation of theoretical knowledge of the discipline;
students gain experience in software development using the C programming language;
acquire skills in basic design, software testing and debugging;
acquire practical skills in software development;
master the skills that will ensure the possibility of high-quality reformatting of the existing software code.*

The subject of the discipline "Fundamentals of Programming. Course work" is a mathematical and algorithmic support of architectural design and software development processes using the C# programming language

*Studying the discipline «Fundamentals of Programming. Course work» generates **general competence (GC) and professional competence (PC):***

***GC 02** Ability to apply knowledge in practical situations.*

***GC 06** Ability to search, process and analyze information from various sources.*

***PC 01** Ability to identify, classify and formulate software requirements.*

***PC 02** Ability to participate in software design, including its structure, behavior and functioning processes modeling (formal description).*

***PC 03** Ability to develop software systems architectures, modules and components.*

PC07 Knowledge of information data models, the ability to create software for data storage, retrieval and processing.

PC 08 Ability to apply fundamental and interdisciplinary knowledge to successfully solve software engineering problems.

PC 10 Ability to accumulate, process and systematize professional knowledge about software creation and maintenance, and determination of the importance of lifelong learning.

PC 11 Ability to implement phases and iterations of the life cycle of the software systems and information technology based on appropriate models and approaches to software development.

PC 13 Ability to reasonably select and master software development and maintenance tools.

PC 14 Ability to algorithmic and logical thinking.

Programming Learning Outcomes (PLO) of the discipline «Fundamentals of Programming. Course work» under the educational program:

PLO 01 To analyze, purposefully search and select the necessary information and reference resources and knowledge to solve professional problems, taking into account modern advances in science and technology.

PLO03 To know the software life cycle basic processes, phases and iterations.

PLO04 To know and to apply professional standards and other normative legal documents in the field of software engineering.

PLO06 Ability to select and use the appropriate task of software development methodology.

PLO07 To know and to apply in practice the fundamental concepts, paradigms and basic principles of the functioning of language, instrumental and computational tools of software engineering.

PLO12 Apply effective approaches to software projection.

PLO13 To know and apply methods of developing algorithms, designing software and data and knowledge structures.

PLO14 To apply in practice the software tools of blast furnace analysis, processing, testing, visualization, measurements and documentation of software

PLO23 To ability to document and present the results of software development.

1. Prerequisites and post-requisites of the course (the place of the course in the scheme of studies in accordance with curriculum)

To the successful study of the discipline "Fundamentals of Programming. Course work" is preceded by the study of the disciplines "Fundamentals of Programming", "Algorithms and Data Structures", "Computer Discrete Mathematics" of the curriculum for bachelors in the specialty 121 Software engineering.

Received during the assimilation of the discipline "Fundamentals of programming. Course work" theoretical knowledge and practical skills ensure successful study of the disciplines "Programming", "Databases", implementation of course projects and master's theses in the specialty 121 Software Engineering.

2. A typical assignment for a term paper

The task of the course work is issued by the teacher - the head of the work and should ensure the conditions of achievement of the educational goal with the possibility of meaningful solution of the tasks. Content of the task

- enter information (book code, author, title, circulation, year of release) in the file

- output all information from the file

- search the file (by key field - book code)

A student can offer a task for the course work, including in view of the possibility of using its results in future scientific work.

The main tasks for the course work are:

- *Creating a new project to enter data into a file*
- *reading data from the file;*
- *recording additional data;*
- *deleting data from the file;*
- *search for information by a given criterion;*
- *drawing up a modular test plan and modular software testing*
- *development of the program and test methods;*
- *conducting tests of software.*

3. Requirements for the structure, content and design of the explanatory note

The results of the course work should be presented in the explanatory note in the form required for assessment of the author's qualification.

Explanatory note of the course work should contain the following sections:

Content;

- 1. Problem setting;*
- 2. Software Specifications (with the submission of a problem solving methods)*
- 3. Testing of software.*
- 4. Program and test method.*
- 5. Test protocol.*

Conclusions.

4. Schedule of course work

- *Analysis of subject area and existing technologies, development of structure of work - until March 15.*
- *Analysis and description of the requirements, development of the user interface - until April 1.*
- *Development of tests, start of development - until April 15.*
- *Demonstration of beta-version of the product-until May 15.*
- *Protection of course work - until May 30.*

5. Coursebooks and teaching resources

- 1. The C Programming Language/ Kernighan,Brian; Ritchie, Dennis M. - Englewood Cliffs, NJ:Prentice Hall, 1988 -288p.*
- 2. C: How to Program/ Paul J. Deitel, Harvey M. Deitel,Pearson Prentice Hall, 2010 -998p.*

Policy and Assessment

6. Course policy

Academic integrity. The policy and principles of academic integrity are defined in section 3 of the Code of Honor of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute". More: <https://kpi.ua/code>.

Norms of ethical behavior. The norms of ethical behavior of students and workers are defined in Section 2 of the Code of Honor of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute". More: <https://kpi.ua/code>.

Inclusive training. The acquisition of knowledge and skills in the course of studying the discipline "Fundamentals of Programming" can be accessible to most persons with special educational needs, except for applicants with serious visual impairments that do not allow tasks using personal computers, laptops and/or other technical means.

Foreign language learning. In the course of tasks, students may be advised to consult English -language sources.

Appointment of incentive and penalty points. Pursuant to the provision on the evaluation system of training results, the sum of all incentive points may not exceed 10% of the rating scale of the evaluation. Attendance at lectures is mandatory.

Attendance at computer lesson can be sporadic and if necessary to protect the work of the computer lesson.

Rules of conduct in the classroom: activity, respect for those present, turning off the phones. Adherence to the policy of academic integrity.

Rules for the protection of computer work: work must be done according to the option of the student, which is determined by his number in the list of the group

7. B Monitoring and grading policy

The maximum number of points for course work: 100 points.

Criteria for quality assessment of software product:

24-25 points-development is done qualitatively, in full;

20-23 points-development is done qualitatively, in full, but has minor disadvantages;

6-19 points-development is sufficient, but contains disadvantages;

0-5 points-development is not fully done or contains significant drawbacks.

Criteria for evaluating planning processes according to the software development methodology:

24-25 points-all tasks planned according to the chosen methodology of software development, the plans were adjusted accordingly;

20-23 points-all tasks planned according to the chosen methodology of software development, plans were not corrected accordingly by changes;

6-19 points-some tasks planned according to the chosen methodology of software development, plans were not corrected accordingly by changes;

0-5 points-tasks not planned according to the selected methodology of software development, plans were not corrected accordingly.

Criteria for evaluating software quality measures:

24-25 points-measures were taken to comply with the proper level of quality product and prevent risks;

20-23 points-measures were taken to comply with the proper level of quality product or prevent risks;

6-19 points-only product testing was carried out;

0-5 points-no measures were taken to comply with the proper quality of the product.

Criteria for assessing the quality and completeness of documentation:

10 points - the documentation is made at a high level, there are no comments;

6-9 points-documentation is done qualitatively, but has disadvantages;

1-5 points-documentation is performed at an acceptable level, but has significant drawbacks;

0 points - documentation is poorly done.

Criteria for assessing the quality and completeness of the presentation and demonstration of the software:

10 points - presentation and demonstration made at a high level, no comments;

6-9 points-presentation and demonstration are made qualitatively, but there are drawbacks;

1-5 points-presentation and demonstration made at an acceptable level, but there are significant drawbacks;

0 points - presentation and demonstration are poorly made.

Criteria for assessing the timeliness of presentation of work for protection:

5 points - work is presented for protection no later than the specified term;

0 points - work is presented for protection later.

The maximum number of points for the performance and protection of course work: 25 points +25 points + 25 points + 10 points + 10 points + 5 points = 100 points.

Semester control: test

Conditions of admission to semester control: with a semester rating at least 30 points and enrollment in all computer works.

The final performance score or the results of the Fail/ Pass Exam are adopted by university grading system as follows:

Score	Grade
100-95	Excellent
94-85	Very good
84-75	Good
74-65	Satisfactory
64-60	Sufficient
Below 60	Fail
Course requirements are not met	Not Graded

Syllabus of the course

Is designed by teacher PhD, Associate Professor, Yuliia Boiarinova

Adopted by Computer Systems Software Department (protocol № 12 from 26.04.23)

Approved by the Faculty Board of Methodology (protocol № 10 from 26.05.23)