Національний технічний університет України
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Department of computer systems software

Programming. Course project

The working program of the academic discipline (Syllabus)

Details of the academic discipline	
Level of higher education	***First (Bachelor)***
Branch of knowledge	*F Information Technologies*
Speciality	*F2 Software engineering*
Educational program	*Software Engineering of Multimedia and Information Retrieval Systems*
Discipline status	*Normative*
Form of education	*Daytime*
Year of training, semester	*2rd year of training, 4th semester*
Scope of the discipline	*Independent work: 60 hours.*
Semester control/ control measures	*Credit*
Lessons schedule	*Not provided*
Language of teaching	*English*
Information about the head of the course/teachers	*Committee for Defense of Course Projects: PhD. , assistant, Artur Oleksii* *oleksii@pzks.kpi.ua*
Placement of the course	Google Classroom access is granted to registered students.

Program of educational discipline

1. Description of the educational discipline, its purpose, subject of study and learning outcomes

The purpose *of completing the course project in the discipline "Programming. Course project" is to consolidate, deepen, and generalize theoretical knowledge regarding methods and technologies of object-oriented programming acquired by students during their studies, as well as their application to the comprehensive solution of a specific professional task, in particular, acquiring skills in building software products (components) for reuse based on design patterns.*

The tasks of this course project are:

– to facilitate students' acquisition of skills in independent work with educational-methodological, technical, monographic, and periodical literature and other information sources;

– to develop the ability to analyze and algorithmize the assigned task, assess the feasibility of applying design patterns for creating software, and justify the chosen structural organization of the project;

– to gain experience in developing software classes (components) suitable for reuse in accordance with the object-oriented approach;

– to develop programming skills, including the creation of software with a complex user interface, and work in programming environments;

– to develop and consolidate the ability to document newly created program code, write technical documentation for the developed project (in particular, technical specifications, descriptions of modules and classes, UML diagrams of patterns, user instructions).

Since the course project is one of the types of individual assignments, its tasks also include the formation of students' skills of independent work in general in educational and professional activities, the ability to take responsibility, independently solve problems, find constructive solutions, overcome

crisis situations, etc. In addition, the completion of the course project contributes to gaining experience in creative and research activities for solving assigned tasks, the ability to develop one's own attitude to solving a practical problem.

***The subject** of the discipline "Programming. Course Project" is the process of developing object-oriented software using design patterns.*

*Studying the discipline "Components of Software Engineering. Course work" helps students to develop the competencies necessary for solving practical tasks of **general (GC) and professional competencies (PC)** related to the design and development of object-oriented software purposes:*

***GC02** Ability to apply knowledge in practical situations.*
***PC01** Ability to identify, classify, and formulate software requirements;*
***PC02** Ability to participate in software design, including modeling (formal description) of its structure, behavior, and functioning processes;*
***PC03** Ability to develop architectures, modules, and components of software systems;*
***PC04** Ability to adhere to specifications, standards, rules, and recommendations in the professional field when implementing life cycle processes;*
***PC07** Possession of knowledge about information models of data, ability to create software for storing, retrieving, and processing data;*
PC08** Ability to apply fundamental and interdisciplinary knowledge for the successful solution of software **engineering tasks;
***PC10** Ability to accumulate, process, and systematize professional knowledge regarding the creation and maintenance of software and recognition of the importance of lifelong learning;*
***PC13** Ability to reasonably choose and master tools for software development and maintenance;*
***PC14** Ability for algorithmic and logical thinking;*
***PC17** Ability to develop software for information-search systems;*
***PC18** Ability to develop software for multimedia, multisemedia, and immersive systems;*
***PC21** Ability to develop scripts for automating two-dimensional and three-dimensional graphic, animation, and descriptive content for import and export of data in multimedia systems;*

*Studying the discipline "Programming. Course project" contributes to students' formation of the following **program learning outcomes (PLO)** according to the educational program:*
***PLO01** Analyze, purposefully search for, and select information-reference resources and knowledge necessary for solving professional tasks, taking into account modern achievements of science and technology;*
***PLO03** Know the main processes, phases, and iterations of the software life cycle;*
***PLO04** Know and apply professional standards and other regulatory legal documents in the field of software engineering;*
***PLO06** Ability to choose and use the appropriate methodology for creating software for the task;*
***PLO07** Know and apply in practice fundamental concepts, paradigms, and basic principles of functioning of language, instrumental, and computational tools of software engineering;*
***PLO08** Ability to develop human-machine interface;*
***PLO09** Conduct pre-project survey of the subject area, system analysis of the design object;*
***PLO12** Apply in practice effective approaches to software design;*
***PLO13** Know and apply methods of algorithm development, software construction, and data and knowledge structures;*
***PLO14** Apply in practice instrumental software tools for domain analysis, design, testing, visualization, measurement, and documentation of software;*
***PLO15** Reasonably choose programming languages and development technologies for solving tasks of creating and maintaining software;*
***PLO16** Have skills in team development, coordination, formatting, and release of all types of software documentation;*
***PLO18** Know and be able to apply information technologies for processing, storing, and transmitting data;*
***PLO23** Be able to document and present the results of software development;*

PLO25 *Know and be able to apply programming technologies for developing software for multimedia and information-search systems;*
PLO30 *Be able to develop scripts for batch import, processing, optimization, and export of multimedia data;*
PLO34 *Know and be able to use basic models of representation of text and multimedia data in the design of information-search and multimedia systems;*
PLO35 *Know and be able to use software tools and technologies for processing multimedia data for developing software for information-search, multimedia, and immersive systems;*
PLO36 *Know and be able to apply principles of building search systems, methods, and algorithms for performing various types of information search in them, criteria for evaluating the effectiveness of information search.*

2. Pre-requisites and post-requisites of the discipline (place in the structural and logical scheme of training according to the relevant educational program)

To the successful study of the discipline " Programming. Course project" precedes the study of the disciplines "Fundamentals of Programming", "Programming", "Components of Software Engineering" of the curriculum of bachelor's training in the specialty F2 Software engineering.
Received during the assimilation of the discipline "Programming. Course projec" theoretical knowledge and practical skills ensure the successful implementation of course projects and bachelor's theses in the specialty F2 Software engineering.

3. Content of the academic discipline

The task of the course work is issued by the teacher - the supervisor of the work and must provide the conditions for achieving the educational goal with the possibility of a meaningful solution to the set tasks. The content of the task is a simple (from the point of view of software development) data processing task, for the solution of which it is necessary to develop software for hands-on task.
The student can propose his own version of the task for the course work, including from the point of view of the possibility of using its results in the future bachelor's thesis.

The main tasks for course project are:
- *Selecting a topic for your course project*
- *Designing a database diagram*
- *Developing the main functionality of the application*
- *Preparing a report on the developed software*
- *Defending the course project*

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

The results of the course work must be presented in an explanatory note in the form necessary for the evaluation of the author's qualifications.
The explanatory note of the coursework should contain the following sections:
- *Table of Contents;*

1. *Introduction;*
2. *Software specifications;*
3. *Software Design and Architecture;*

3.1. Database Design;
3.2. Application Architecture;

4. *Implementation and Functionality;*
5. *Conclusions;*
6. *References;*
7. *Appendix.*

5. Course work schedule

- *Analysis of the subject area and available technologies, development of conceptual architecture - by March 15.*
- *App and database designing - by April 1.*
- *Development of of the main functionality - by May 1.*
- *Demonstration of the beta version of the product - until May 15.*
- *Defence of course work - until May 30.*

6. Methods of mastering an educational discipline (educational component)

Basic literature:

1. *Educational materials from the discipline "Programming. Course work". Use to master practical skills in the discipline. The materials are in Google classroom. Access is granted to registered students.*

Policy and control

7. The policy of academic discipline (educational component)

- *Adherence to the policy of academic integrity.*
- *Rules for protecting the work of the computer workshop: the work must be done according to the student's option, which is determined by his number in the group list.*
- *The rules for assigning incentive and penalty points are as follows.*

Penalty points are calculated for:

- plagiarism (the program code does not correspond to the task variant, the identity of the program code among different works): -15 points

.8. Types of control and rating system for evaluating learning outcomes (RSO)

The maximum number of points for the course work: 100 points.
Criteria for evaluating the quality of a software product:
24-25 points – the development is done qualitatively, in full;
20-23 points – the development is done qualitatively, in full, but has minor flaws;
6-19 points – the development is carried out to a sufficient extent, but contains shortcomings;
0-5 points – the development is not completed in full or contains significant shortcomings.

Criteria for evaluating compliance with planning processes according to the software development methodology:
24-25 points – all tasks are planned according to the selected software development methodology, plans were adjusted according to changes;
20-23 points – all tasks are planned according to the selected software development methodology, plans were not adjusted according to changes;
6-19 points – some tasks are planned according to the selected software development methodology, plans were not adjusted according to changes;
0-5 points – tasks were not planned according to the selected software development methodology, plans were not adjusted according to changes.

Criteria for OOP implementation quality and robustness:
24–25 points – high-quality OOP implementation and robust code;
20-23 points – good OOP implementation with minor flaws;

6-19 points – some OOP effort, but weak quality assurance;
0-5 points – no meaningful OOP or quality assurance.

Criteria for evaluating the quality and completeness of documentation:
10 points – the documentation is done at a high level, there are no comments;
6-9 points – the documentation is done qualitatively, but has shortcomings;
1-5 points – the documentation is completed at an acceptable level, but has significant shortcomings;
0 points - the documentation is done poorly.

Criteria for evaluating the quality and completeness of the presentation and demonstration of the software product:
10 points – the presentation and demonstration were performed at a high level, there are no comments;
6-9 points – the presentation and demonstration are done well, but there are shortcomings;
1-5 points – the presentation and demonstration are performed at an acceptable level, but there are significant shortcomings;
0 points – the presentation and demonstration were performed poorly.

Criteria for evaluating the timeliness of work submission for defense:
5 points – the work is submitted for defense no later than the specified deadline;
0 points – the work is submitted for defense later than the specified deadline.

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

Semester control: assessment.

Scores	*Rating*
100-95	Perfectly
94-85	Very good
84-75	Fine
74-65	Satisfactorily
64-60	Enough
Less than 60	Unsatisfactorily
Admission conditions not met	Not allowed

The working program of the academic discipline (syllabus):

Compiled by PhD, assistant Artur Oleksii;

Adopted by Computer Systems Software Department (protocol № 15 from 15.04.26)

Approved by the Faculty Board of Methodology (protocol № 11 from 22.05.26)