Software Engineering Components. Coursework.

The program of the academic discipline (Syllabus)

Details of the academic discipline

Cycle of Higher Education	*First cycle of higher education (Bachelor's degree)*
Field of Study	*12 Information technologies*
Specialty	*121 Software Engineering*
Education Program	*Computer Systems Software Engineering*
Type of Course	*Normative*
Mode of Studies	*Full-time education*
Year of studies, semester	*3 year (5 semester)*
ECTS workload	*1 credit (30 hours)*
Testing and assessment	*Test*
Course Schedule	*https://roz.kpi.ua*
Language of Instruction	*English*
Course Instructors	Prof., Dr.Sc. Bogdan Korniyenko, b.korniyenko@kpi.ua
Access to the course	https://classroom.google.com

Program of academic discipline

1 Course description, goals and objectives, and learning outcomes

Coursework is performed with the aim of consolidating, deepening and generalizing the knowledge acquired by students during their studies and their application to the complex solution of a specific professional task. Writing and defending a term paper is an important preparatory stage for the implementation of the next, more difficult task - the completion of bachelor's and master's theses. Course work on the components of software engineering is a mandatory component of the educational and professional program "Computer systems software engineering" for obtaining a bachelor's degree.When studying this discipline, students will acquire practical software development skills.

The purpose of the course work is to systematize, consolidate and deepen the theoretical and practical knowledge obtained during the study of the discipline "Components of software engineering", the formation of skills in the application of this knowledge when solving specific practical problems in the subject area of computer science.

The subject of study of the discipline is theory and practice of applying basic algorithmic structures and basic data structures using modern software development technologies.

According to the requirements of the EP, the discipline "Software Engineering Components" should ensure that applicants acquire competencies and program learning outcomes: PC01-05, PC07, PC08, PC11-13, PLO01-04, PLO06-11, PLO13-20. In particular, after mastering the module "Software Engineering Components. Coursework" must demonstrate the following competencies and program learning outcomes:

- the ability to analyze subject areas, form, classify software requirements;

- the ability to develop and implement scientific and/or applied projects in the field of software engineering;
- the ability to design software architecture, to model the functioning processes of individual subsystems and modules;
- the ability to develop and implement new competitive ideas in software engineering;
- the ability to develop, analyze and apply specifications, standards, rules and recommendations in the field of software engineering;
- the ability to critically consider problems in the field of information technology and at the border of the fields of knowledge, integrate relevant knowledge and solve complex problems in broad or multidisciplinary contexts;
- the ability to develop and coordinate processes, stages and iterations of the software life cycle based on the application of modern software development models, methods and technologies;
- ability to ensure software quality;
- ability to develop software using concepts of information security;
- the ability to use technologies that ensure the integrity of databases and network security;
- the ability to generalize the results of scientific and project activities;
- the ability to use the latest approaches in the field of information technologies to process data obtained in conditions of interval uncertainty.

According to the results of studying the educational discipline "Software Engineering Components. Coursework", the following **knowledge** should be obtained:

- in-depth study of the principles of structural programming, modern procedural and object-oriented languages, basic data structures and the ability to apply them during the software implementation of algorithms for professional tasks;
- apply modern technologies and tools for the development of software systems at all stages of the life cycle;
- explain the difference between different programming paradigms, characterize types of programming, classify software development methods;
- identify, classify and formulate software requirements;
- formulate and ensure software quality requirements in accordance with customer requirements, specifications and standards;
- the ability to evaluate and take into account economic, social, technological and environmental factors affecting the field of professional activity.

Skills that should be acquired as part of studying the academic discipline "Software Engineering Components. Coursework":

- acquisition of practical skills in developing programs using various programming paradigms: generalized, object-oriented, functional, logical, with appropriate models, calculation methods and algorithms, data structures and control mechanisms;
- acquisition of practical skills for solving various problems in programming processes and objects;
- acquisition of practical skills regarding the justification of the choice of the development environment;
- implementation in the form of a program of one or more interconnected algorithms that solve the given applied problem;
- application of basic regulatory documents necessary for designing, developing and designing software products.

Such a combination of general and special competences, theoretical and practical knowledge, skills and abilities helps to increase the professional level of bachelor's degree holders in order to carry out effective activities in the field of development ofsoftware engineering.

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

Necessary disciplines: "Algorithms and data structures", "Fundamentals of programming", "Software Engineering Components. Part 1. Introduction to software engineering", "Software Engineering Components. Part 2. Software modeling. Analysis of requirements for software", "Software Engineering Components. Part 3. Software architecture", "Software Engineering Components. Part 4. Software quality and testing", “Object-oriented programming”.

Module "Software Engineering Components. Coursework." is necessary for studying the disciplines "Risk management and project quality", “Complex Systems Design”, “Diploma Design”.

3 Structure of the credit module

The course work is an individual task from the discipline "Software Engineering Components" and is being prepared for defense in the final period of theoretical training. The coursework must be prepared for defense within the time limit set by the teacher. An explanatory note is submitted to the defense of the coursework.

The explanatory note includes the following components: title page, term paper task, table of contents, which includes the names of all sections and points with page numbers, introduction, which indicates the purpose and task of the term paper; the theoretical part, which describes theoretical information on the topic of the work; practical implementation in the programming language. At the end of the explanatory note, a conclusion based on the results of the work is presented.

4 Educational resources and materials

Basic:

1. Pressman, Roger (2010) *Software Engineering: A Practitioner's Approach*, McGraw Hill, New York, NY.
2. Sommerville, Ian (2011) *Software Engineering*, Addison-Wesley, Boston, MA.
3. Stephens, Rod (2015) *Beginning Software Engineering*, Wrox.
4. Tsui, Frank , Orlando Karam and Barbara Bernal (2013) *Essentials of Software Engineering*, Jones & Bartlett Learning , Sudbury, MA.
5. Pfleeger, Shari (2001) *Sofwtare Engineering: Theory and Practice*, Prentice Hall, Upper Saddle River, NJ.

Supplementary:

1 Larman, C. (2005) *Applying UML and Patterns: AnIntroduction to Object-Oriented Analysis and Design and iterative Development*, Pearson

2 Ambler, S. (2002) *Agile Modeling: Effective Practices for Extreme Programming and the Unified Process*, NewYork, John Wiley & Sons.

3 Bass, D.L., Clements, D.P. andKazman, D.R. (2012) *Software Architecture in Practice*, 3rd edn, Upper Saddle River, NJ, Addison Wesley

4 Beck, K. (2004) *Extreme Programming Explained: Embrace Change*, Upper Saddle River, NJ, Addison Wesley

5 Clemens Szyperski (2002) Component Software: Beyond object-oriented programming, Addison-Wesley

6 John Cheesman & John Daniels (2000) UML Components: A simple process for specifying component-based software (The component software series) Addison-Wesley

7 Rob Pooley, Perdita Stevens (2006) *Using UML Software Engineering with Objects and Components*, second edition. Addison-Wesley

8 Christopher Fox (2006) *Introduction to Software Engineering Design*. Addison Wesley

Attendance Policy and Assessment

5 Attendance Policy

During classes in an academic discipline, students must adhere to certain disciplinary rules:
• extraneous conversations or other noise that interferes with classes are not allowed;
• the use of mobile phones and other technical means is not allowed without the teacher's permission.

The coursework is submitted (defended) personally with verification of the obtained practical results and theoretical knowledge necessary for the performance of such work. Validation of practical results includes code review and execution of test tasks.

6 Monitoring and grading policy

Semester control: test

Admission of a student of higher education to the defense of a course work is carried out by the academic supervisor. The admission criteria are:

- availability of an electronic version of the text part of the coursework in .doc or .docx format designed according to the requirements;
- the presence of an electronic version of a working software development (according to the task set in the work), presented in the form of an installer for one or more of the common modern operating systems;
- available bound printed copy of the text part of the course work, designed according to the requirements, endorsed by the supervisor;
- compliance of the content of the text part with the topic of the course work;
- presence in the appendices of the text part of the coursework of the technical task and user instructions for using the software development;
- observance of academic integrity during coursework writing, in accordance with regulatory documents.

The coursework defense includes a short speech by the student with a presentation, his answers to the questions of the committee members. The student's presentation reflects the relevance of the topic, coursework tasks, its main results and a demonstration of the software product. The student must demonstrate the ability to answer questions from the subject area of the course work, conduct a scientific discussion.
At the end of the defense procedure, the commission makes a decision on the final overall grade for the course work.

Table1 — Correspondence of rating points to grades on the university scale

Score	*Grade*
100-95	Exellent
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:

designed by Professor of the Department of Information Systems and Technologies, Bogdan Korniyenko

adopted by Department of Computer Engineering (protocol № 10, 25.05.2022)

approved by the methodical commission of FICT (protocol № 10, 09.06.2022)