National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute"

Department of Computer Engineering

Programming Fundamentals. Coursework

Syllabus

Requisites of the Course

Level of Higher Education	***First (undergraduate)***
Field of Study	*12 Information technologies*
Speciality	*121 Software engineering*
Education Program	*Software engineering of computer systems*
Type of Course	*Normative*
Mode of Studies	*full-time*
Year of studies, semester	*1st year, spring semester*
ECTS workload	*1 credit (30 hours)*
Testing and assessment	*Test*
Course Schedule	*http://roz.kpi.ua/*
Language of Instruction	*English*
***Head of the course /* Course Instructors**	*Head of the course: head of the department, D.sc., Prof. Stirenko S. G. , sergii.stirenko@gmail.com* *Shemsedinov T.G., senior lecturer of the Department of Computer Engineering. timur _ shemsedinov @ gmail . com*
Access to the course	https://github.com/HowProgrammingWorks

Outline of the Course

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

The purpose of studying the credit module " Basics of programming. Coursework » is the consolidation, deepening, and generalization of theoretical knowledge and practical skills that students acquire while completing tasks on the basics of programming. According to the results of studying the discipline, the student should be able to solve professional tasks and possess the following competencies:

- Ability to abstract thinking, analysis and synthesis (GC01)
- Ability to search, process and analyze information from various sources (GC06)
- Ability to identify, classify and formulate Software requirements (PC01)
  Ability to participate in Software Design, including Modeling (formal description) its Structure, Behavior, and Operating Processes (PC02)
- Ability to develop Architectures, Modules and Program System Components (PC03)

- Knowledge of Information Data Models, ability to create Software for storing, extracting and processing Data (PC07)
- Ability to use Fundamental and Interdisciplinary Knowledge to successfully solve Software Engineering problems (PC08)
- Ability to accumulate, process and systematize Professional Knowledge about the creation and maintenance of Software and recognition of the importance of lifelong learning (PC10)
- Ability to Algorithmic and Logic thinking (PC14)

After mastering the academic discipline, students must demonstrate the following program learning outcomes:

- Analyze, purposefully search and select the Information and Reference Resources and Knowledge necessary for solving Professional Tasks, taking into account the Modern Achievements of Science and Technology (PLO01)
- Know the basic Processes, Phases, and Iterations of the Software Lifecycle (PLO03)
- Know and apply in practice the Fundamental Concepts, Paradigms and Basic Principles of functioning of Language, Instrumental and Computational Means of Software Engineering (PLO07)
- Know and apply methods for developing Algorithms, Software Design and Data and Knowledge Structures (PLO13)
- Know and be able to apply Information Technologies for Data Processing, Storage and Transmission (PLO18)
- Be able to document and present Software Development Results (PLO23)

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

Preceding disciplines: Programming Fundamentals. Part 1

Disciplines for which this course prepares: Components of software engineering. Parts 1 and 2, Fundamentals of Node.Js Software Development, Systems Programming, Parallel and Distributed Computing, Agile Programming Techniques, Software Modeling. .

3. Content of the academic discipline

Topic 1. State of applications, data structures and collections
Topic 2. Approaches to working with state: statefulandstateless
Topic 3. Structures and records
Topic 4. Stack, queue, dec
Topic 5. Trees and graphs
Topic 6. Projections and display of data sets
Topic 7. Estimation of computational complexity
Topic 8. Structure of the application: files, modules, components
Topic 9. Object, prototype and class
Topic 10. Dependencies and libraries
Topic 11. Regular expressions

Topic 12. Factories and pools
Topic 13. I/O (input-output) and files
Topic 14. Monomorphic and polymorphic code, inline cache, hidden classes
Topic 15. Code performance measurement and optimization
Topic 16. Asynchronous programming on callbacks
Topic 17. Asynchronous programming on promises
Topic 18. Asynchronous functions, async/await, thenable, error handling
Topic 19. Immutable data structures
Topic 20. Automatic programming: finite state machines (state machines)
Topic 21. JavaScript Singleton template
Topic 22. Functional objects, functors and monads
Topic 23. Asynchronous generators and asynchronous iterators
Topic 24. Enumerated type (enum)

Stages of course work

- The main stages of course work:
- Getting a topic and task
- Selection and study of literature
- Formation of the technical task
- Development of coursework sections and software application
- Software application testing
- Issuance of an explanatory note
- Submission of a course project (work) for review
- Protection of course project (work)

4. Educational materials and resources

Basic:

1. Shemsedinov T.G., Nechay D.O., Kuhar V.V., Orlenko O.A., Golikov O.G., Bilochub M.M., Dukhin V., Ivanova L.A., Chornenkyi A.Yu. . and other. Code examples and project examples [Electronic resource] are available at: https://github.com/HowProgrammingWorks/
2. Refactoring: Improving the Design of Existing Code // MartinFowler
3. Clean Code: A Handbook of Agile Software Craftsmanship // Robert C. Martin
4. Introduction to Algorithms, 3rd Edition // Thomas H. Cormen
5. Design Patterns // Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides

Additional:

1. Shemsedinov T.G., Nechay D.O., Kuhar V.V., Orlenko O.A., Golikov O.G., Bilochub M.M., Dukhin V., Ivanova L.A., Chornenkyi A.Yu. . and other. The Metarhia technology stack [Electronic resource] is located at: https://github.com/metarhia/
2. Algorithms Unlocked // Thomas H. Cormen
3. The Art of Computer Programming // Donald Knuth
4. Code Complete // Steve McConnell

5. Designing Object Oriented C++ Applications Using TheBooch Method // Robert C. Martin
6. Extreme Programming Explained // Kent Beck
7. Analysis Patterns: Reusable Object Models // Martin Fowler

Educational content

5. Methodology

Coursework consists of nine stages, which are listed in following table

Semester week	The name of the stage of work
3	Getting a topic and task
4-5	Selection and study of literature
6-7	Formation of the technical task
8-13	Development of coursework sections and software application
14	Software application testing
15	Issuance of an explanatory note
16	Submission of a course project (work) for review
17	Protection of course project (work)

6. Self-study

No s/p	**The name of the topic submitted for independent processing**	**Number of hours**
1	Obtaining a topic and assignment for a term paper	1
2	Analysis of the task, selection and study of the literature	5
3	Development of coursework sections and the software itself Appendices	19
4	Drawing up an explanatory note to the term paper	5
	Total	30

Educational content

7. Course Policy

The design of the coursework must meet the requirements for reports on the National Development and Reform Commission (DSTU 3008-2015 "State Standard of Ukraine. Documentation. Reports in the field of science and technology. Structure and rules of design").

All illustrative material in the course work must be completed with the help of computer tools. The content of the illustrative material must adequately reflect the main provisions that are being defended.

Both the teacher and the student are obliged to adhere to the Code of Honor of the National Technical University of Ukraine "Kyiv Polytechnic Institute named after Igor Sikorsky".

The main provisions of the policy:

- the topic of the course work can be coordinated with the topic of the future qualification work of the bachelor;

- stages of the course work must be completed according to the established calendar work schedule;

- the developed software application must be tested, the results of testing the software application are given in the text of the main part of the course work;

- in the case of detection of academic dishonesty and plagiarism, the coursework is returned for thorough revision with a possible change of topic;

- untimely completion of the coursework stage entails a 10% reduction of the points received for it, if the delay is no more than two weeks, and 20% if the delay is more than two weeks.

The following factors are taken into account when evaluating course work:

• complete completion of an individual coursework assignment;
• correctness of developed precedents;
• timeliness of course work according to the schedule;
• independent performance of coursework and absence of signs of plagiarism;
• answers to questions about the content of the coursework during its defense.

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

The system for evaluating the success of students in the discipline "Fundamentals of programming. The coursework" is based on the "Regulations on the system of evaluation of learning results in KPI named after Igor Sikorskyi" (**https://document.kpi.ua/files/2020_1-273.pdf**), the second type of rating evaluation system (RSO-2). RSO-2 coursework (**RK**) consists of two components :

• start (**RS**);
• component of protection (**RZ**).

RK = RS + RZ

The first (initial) component characterizes the student's course work and its result - the quality of the explanatory note and the developed software application. The secondary syllable characterizes the quality of the student's defense of the course work.

The size of the scale of the first component equals **80 points** , and the second component - **20 points** . **The quality of the explanatory note and the degree of compliance with the calendar schedule of work** Weighted score **- 80** (RS). The criteria for evaluating the components of the explanatory note are listed in Table 8.1.

Since the credit module has a semester certification in the form of credit, the rating evaluation system is built according to the RSO type - 1. The rating of the student from the credit module consists of the points he receives for the types of work according to table.

Evaluation criteria for the implementation of the components of the explanatory note

Stage No	**Composite works**	**The maximum number of points for timely performance**	**Taking into account the timeliness of execution**
1	Layout of the title page	2	**100%** of the grade if the
2	Availability of a technical task for the KR	2	
3	Availability and content of the album description	2	

4	Availability of content	2	work schedule is followed **90%** in case of delay **up to 2 weeks** **80%** in case of a delay **of more than 2 weeks**
5	Presence and contents of the stupa	2	
6	The availability and content of the development of all sections of the term paper assignment.	45	
7	Availability and content of the software application	10	
8	Availability and content of software application testing results	5	
9	Availability and content of conclusions	5	
10	Availability and registration of the list of sources	5	
	In just one semester	80	
	Protection of term paper	20	
	That's all	**100**	

A student is allowed to defend a coursework on the condition that he has an initial RS component of at least 60% of the maximum value, which is **80 x 0.6 = 48 points.**

Quality of protection

Weight score – **20** (RZ).

Evaluation criteria for the performance with a report based on the materials of the KR and answers to

question:

– mastery of theoretical material up to 10 points;

– the degree of mastery of software application development methods in total up to 10 points.

The defense of the coursework is considered successful if the RZ is at least 60% of its maximum value, i.e. **20 x 0.6 = 12 points.**

After the defense of the coursework is completed, **the RK is determined** , which is later translated into an assessment on the university scale according to the table:

Correspondence of rating points to grades on the university scale

Rating	*Grade*
100-95	Excelent
94-85	Very good
84-75	Good
74-65	Satisfactorily
64-60	Sufficient
Less than 60	Fail
Admission conditions not met	Not Graded

Working program of the academic discipline (syllabus):

designed by Shemsedinov T.G., a senior lecturer at the Department of Computer Engineering

adopted by the Department of Computer Engineering (Protocol No. 10 dated 05/25/2022)

agreed by the Methodical Commission of the faculty (protocol No. 10 dated 06/09/2022)