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

Computer Systems Software Department

ABSTRACT ALGEBRA FOR SOFTWARE ENGINEERING

Working program of the academic discipline (Syllabus)

Details of the academic discipline	
Level of Higher Education	***First (bachelor's)***
Field of Study	*12 Information Technologies*
Specialty	*121 Software Engineering*
Education Program	*Software Engineering of Multimedia and Information Retrieval Systems*
Type of Course	*Selective*
Mode of Studies	*full-time*
Year of studies, semester	*3rd year, 6th semester*
ECTS workload	*Lectures: 36 hours, computer workshop: 18 hours, independent work: 66 hours.*
Testing and assessment	*Assessment, modular control work, calendar control*
Course Schedule	*According to the schedule for the spring semester of the current academic year (rozklad.kpi.ua)*
Language of Instruction	*English*
Course Instructors	Lecturer: *Ph.D., Associate Professor, Onai Mykola* Practical training: *Ph.D., Associate Professor, Onai Mykola*

Outline of the Course

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

Studying the discipline "Abstract Algebra for Software Engineering" allows students of higher education to develop the competencies necessary for solving complex problems of professional activity related to the development of the mathematical basis of software information security systems.

***The purpose** of studying the discipline "Abstract Algebra for Software Engineering" is to form a competent specialist in the field of abstract algebra for software engineering, able to apply and develop the basic principles and methods of the discipline in production activities, independently analyze the structure of algebraic objects, build mathematical models, apply apparatus of the discipline for the study of abstract algebraic structures. Important tasks are the formation of algebraic and theoretical-numerical culture in students, the promotion of the development of logical and analytical thinking of students, the provision of information to students about the directions of development of modern mathematics, in particular, mathematics that is used in cryptographic methods of information protection.*

***The subject of** the discipline "Abstract Algebra for Software Engineering" is the methods and algorithms of abstract algebra and number theory when solving information protection problems using software tools.*

*The study of the discipline "Abstract Algebra for Software Engineering" strengthens the formation of students **of professional competences (PC)** necessary for solving practical problems of professional activity:*
***PC18** Ability to apply the acquired fundamental mathematical knowledge for the development of calculation methods in the creation of multimedia and information retrieval systems.*

*The study of the discipline "Abstract Algebra for Software Engineering" contributes to the formation of the following **learning outcomes** for students according to the educational program:*
***PLO01** To analyze, purposefully search for and select for the information and reference resources and knowledge necessary for solving professional tasks, taking into account modern achievements of science and technology.*

***PLO25** To know and to be able to use fundamental mathematical tools to build algorithms and develop modern software.*

***PLO26** To be able to develop and use methods and algorithms for the approximate solution of mathematical problems in the design of multimedia and information retrieval systems.*

2. Prerequisites and post-requisites of the course (the place of the course in the structural-logical scheme of studies in accordance with educational program)

The successful study of the discipline "Abstract Algebra for Software Engineering" is preceded by the study of the disciplines "Algorithms and data structures" and "Algorithmic support of multimedia and information-search systems" of the bachelor's training plan for the specialty 121 Software Engineering.

The theoretical knowledge and practical skills obtained as a result of mastering the discipline "Abstract Algebra for Software Engineering" can be useful for conducting scientific research and completing bachelor's qualification work.

3. Content of the course

The discipline "Abstract Algebra for Software Engineering" involves the study of topics:

Topic 1. Basic provisions of abstract algebra

Topic 2. Algebraic structures used in computer algebra

Topic 3. Simple numbers in software engineering

Topic 4. Cryptography based on abstract algebra

Topic 5. Interference-resistant coding based on abstract algebra

Modular control work

Test

4. Educational materials and resources

Basic literature:

1. *Cyber security: modern protection technologies. Study guide for students of higher educational institutions. / S.E. Ostapov, S.P. Yevseev, O.G. King. – Lviv: "New World-2000", 2020. - 678 p*
2. *DSTU ISO/IEC 27001:2023 Informatsiina bezpeka, kiberbezpeka ta zakhyst konfidentsiinosti. Systemy keruvannia informatsiinoiu bezpekoiu. Vymohy (ISO/IEC 27001:2022, IDT)*
3. *Tekhnolohii zakhystu informatsii [Elektronnyi resurs] : pidruchnyk dlia stud. spetsialnosti 122 «Kompiuterni nauky», spetsializatsii «Informatsiini tekhnolohii monitorynhu dovkillia», «Heometrychne modeliuvannia v informatsiinykh systemakh» / Yu. A. Tarnavskyi; KPI im. Ihoria Sikorskoho. – Elektronni tekstovi dani (1 fail: 2,04 Mbait). – Kyiv : KPI im. Ihoria Sikorskoho, 2018. – 162 s.*
4. *Volodymyr Stepanovych Blintsov, Yury Leonidovych Galchevskyi. Mathematical foundations of cryptology: Study guide for students. higher education closing / National University of Shipbuilding named after Admiral Makarov. - Mykolaiv: NUK, 2006. - 232p. : fig., table; - ISBN 966-321-056-*
5. *State standard of Ukraine. Information Technology. Cryptographic protection of information. A digital signature based on elliptic curves. Formation and verification. DSTU 4145-2002*

Additional literature:

1. *Jean-Philippe Aumasson Serious Cryptography: A Practical Introduction to Modern Encryption, 2-nd Edition : No Starch Press, 2024*
2. *Michael E. Wihitman, Herbert J. Mattord Principles of Information Security, Sixth Edition, Kennesaw State University.*
3. *Wenbo Mao Modern Cryptography: Theory and Practice : Pearson P T R; 1st edition*
4. *Thomas R. Shemanske Modern Cryptography and Elliptic Curves: A Beginner's Guide : American Mathematical Society (July 31, 2017)*
5. *Niels Ferguson, Bruce Schneier, Tadayoshi Kohno Cryptography Engineering: Design Principles and Practical Applications Wiley*
6. *Seth James Nielson, Christopher K. Monson Practical Cryptography in Python: Learning Correct Cryptography by Example Apress; 1st ed. edition (September 27, 2019)*
7. *Lawrence C. Washington Elliptic Curves: Number Theory and Cryptography Chapman and Hall/CRC; 2nd edition*

Educational content

5. Methodology of mastering the discipline (educational component)

No.	*Type of training session*	*Description of the training session*
Topic 1. Basic provisions of abstract algebra		
1	*Lecture 1. Basic algebraic structures*	*Definition of binary operation of algebra. Algebraic structures with one binary operation. Group concept. Examples and properties of groups. Subgroups. Normal subgroups and factor groups. Homomorphisms of groups. Isomorphism. Algebraic structures with two binary algebraic operations. Ring concept. Examples and properties of rings. Subrings Ring ideals. Ring factor.*
2	*Lecture 2. General provisions of the computer algebra system*	*General formulation of problems of analytical transformations using a computer. Analytical transformations using a computer. Efficiency of algorithms. Presentation of data in the computer (numbers, fractions, polynomials, functions, matrices, series)*
3	*Computer workshop 1*	*Develop a program in any programming language that will implement basic unary operations in algebraic structures*
Topic 2. Algebraic structures used in computer algebra		
4	*Lecture 3. Ring of integers. Theory of divisibility in the ring of integers*	*A ring of integers. The divisibility ratio, its simplest properties. Division with Remainder Theorem. A ring of classes of deductions. NSD, NSK: Euclid's algorithm and Lamé's theorem; extended Euclid algorithm; Euclid's algorithm and continued fractions. Simple numbers. Decomposition of whole numbers into factors; factoring large integers. Exact calculations using modular arithmetic. Representation of large integers in*

		computer memory. Extracting roots from large integers. Checking properties of large integers.
5	*Lecture 4. Ring of polynomials in one variable*	*Construction of rings of polynomials over the field. Divisibility of polynomials. Divisibility theorem with a remainder. Binomial divisibility, Horner's scheme, Taylor's formula. Roots of a polynomial, Bezou's theorem. NSD and NSC polynomials. Euclid's algorithm and its consequences. Mutually simple polynomials. Reducible and irreducible polynomials. Expansion into irreducible factors, unity of expansion. The concept of polynomials in several variables.*
6	*Computer workshop 2*	*Develop a program in any programming language that will implement basic binary operations in algebraic structures*
7	*Lecture 5. Field expansion. Formal integration*	*Algebraic expansion of the field. Formulation of problems of formal integration. Integration of optimal functions. Finite field extensions. Finite fields*
8	*Lecture 6. Arithmetic in classes of modulo remainders, Fermat's and Euler's Theorem*	*The concept of a ring is gone. Structure of the multiplicative group of the ring of remainders modulo. The homomorphic image of the ring of integers.*
9	*Computer workshop 3*	*Develop a program in any programming language that will implement basic binary operations in algebraic fields*
Topic 3. Simple numbers in software engineering		
10	*Lecture 7. Algorithms for checking numbers for simplicity and arithmetic of long numbers*	*Naive methods. Probability tests. Deterministic tests. Miller-Rabin test. Solovei-Strassen test. AKS test.*
11	*Lecture 8. Generation of random prime numbers*	*Linear congruent method and its modifications. John Mockley's method. Additive method. Lagging Fibonacci methods. Polynomial method. Inverse congruent method. Combined methods. A method of generating a binary sequence*
12	*Computer workshop 4*	*Develop a program in any programming language that will implement simplicity testing of long integers*
13	*Modular control work 1. Part 1*	
Topic 4. Cryptography based on abstract algebra		
14	*Lecture 9. Principles of building cryptosystems with a public key*	*One-sided function. Key distribution protocols.*
15	*Lecture 10. El-Gamal scheme and methods of finding a discrete logarithm*	*Selection of El-Gamal scheme keys. Open and secret parameters of the El-Gamal scheme. Encryption and decryption algorithm according to the El-Gamal*

		method. Element index by module. Index existence condition. One-sided function. Discrete logarithmization. Shanks method.
16	*Computer workshop 5*	*Develop a program in any programming language that will implement discrete logarithms*
17	*Lecture 11. Definition of the elliptic curve concept*	*An elliptic curve in Weierstrass form. Elementary operations on points of an elliptic curve. Projective coordinate system. Singular elliptic curves.*
18	*Lecture 12. Finite fields GF(p)*	*Varieties of Galois fields. Multiplicative group of a finite field. The order of the multiplicative group of the remainder ring. Finding the multiplicative inverse element in the field GF(p).*
19	*Lecture 13. Finite fields GF(p^m)*	*Peculiarities of performing operations on the elements of the field GF(p^m). Ways of specifying the elements of the field GF(p^m).*
20	*Computer workshop 6*	*Develop a program in any programming language that will implement basic binary operations on finite fields*
21	*Lecture 14. Elliptical analogue of key exchange according to the Diffie-Hellman scheme*	*The main mathematical operation in the Diffie-Hellman scheme on an elliptic curve. Multiplying a point on an elliptic curve by a number. Connection of the discrete logarithm problem on an elliptic curve with the Diffie-Hellman algorithm. Generalization of the Diffie-Hellman algorithm in elliptic cryptography.*
22	*Computer workshop 7*	*Develop a program in any programming language that will implement the Diffie-Hellman scheme*
Topic 5. Interference-resistant coding based on abstract algebra		
23	*Lecture 15. Elements of the theory of interference-resistant coding*	*Principles of error detection and correction in tamper-resistant codes. Classification and main characteristics of interference-resistant codes. Hamming code*
24	*Computer workshop 8*	*Develop a program in any programming language that will implement the Hamming code*
25	*Lecture 16. BCH codes and the Reed-Solomon code*	*Bowes-Choudhury-Hawkingham code. Reed-Solomon code*
26	*Computer workshop 9*	*Summing up*
27	*Modular control work. Part 2*	

6. Independent work of a student/graduate student

The discipline "Abstract Algebra for Software Engineering" is based on independent preparation for classroom classes on theoretical and practical topics.

No. z/p	*The name of the topic submitted for independent processing*	*Number of hours*	*literature*
1	*Preparation for lectures*	*16*	*1-5*

2	*Preparation for a computer workshop*	*27*	*1-5*
3	*Preparation for modular control work. Part 1*	*9*	*1-3*
4	*Preparation for modular control work. Part 2*	*9*	*2-5*
5	*Preparation for the test*	*5*	*1-5*

Policy and Assessment

7. Policy of academic discipline (educational component)

Attending classes. *Absence from a classroom session does not involve the calculation of penalty points, since the student's final rating score is formed solely on the basis of the evaluation of study results. At the same time, discussion of the results of the thematic tasks, as well as presentation / public speaking and participation in discussions and additions at seminars will be evaluated during classroom classes. In order to actively participate in the work of the seminar, the student prepares for a specific seminar class in literature as recommended by the teacher. Participation in the work of the seminar also involves the preparation of reports and co-reports within all classes.*

Missed evaluation control measures. *Every student has the right to make up lessons missed for a valid reason (hospital, mobility, etc.) at the expense of independent work. More details at the link: https://kpi.ua/files/n3277.pdf .*

The procedure for contesting the results of assessment control measures. *A student may raise any issue relating to the assessment procedure and expect it to be dealt with in accordance with pre-defined procedures. Students have the right to challenge the results of control measures with arguments, explaining which criteria they disagree with according to the evaluation. Calendar control is carried out in order to improve the quality of students' education and monitor the student's fulfillment of the syllabus requirements.*

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

Norms of ethical behavior. *Standards of ethical behavior of students and employees are defined in Chapter 2 of the Code of Honor of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute". More details: https://kpi.ua/code .*

Inclusive education. *The acquisition of knowledge and skills in the course of studying the discipline "Research activity in computer engineering" can be accessible to most people with special educational needs, except for students with serious visual impairments that do not allow them to perform tasks with the help of personal computers, laptops and/or other technical means.*

Studying in a foreign language. *In the course of the tasks, students may be recommended to refer to English-language sources. Assigning incentive and penalty points According to the Regulation on the system of evaluation of learning results, the sum of all incentive points cannot exceed 10% of the rating scale.*

All students must attend lectures and practical classes, where you need to actively work on learning the learning material. For objective reasons (for example - illness, international internship), training can take place in an online form individually upon agreement with the head of the course.

Deadlines and Rescheduling Policy:

Works that are submitted late without good reason will be assigned a lower grade. Rearranging modules takes place with the permission of the dean's office if there are good reasons (for example, sick leave).

Policy on academic integrity *:*

All written works are checked for plagiarism and accepted for defense with correct textual borrowings of no more than 20%. Write-offs during control work are prohibited (including using mobile devices).

8. Types of control and rating system of assessment of learning outcomes

During the semester, students perform 8 computer workshops. The maximum number of points for each computer workshop: 6 points.

Points are awarded for:
- quality of performance of the computer workshop: 0-2 points;
- answer to theoretical questions during the defense of the computer workshop: 0-2 points;
- timely presentation of work for defense: 0-2 points.

Performance evaluation criteria:
2 points – the work is done qualitatively, in full;
1 point - the work is completed in full, but contains minor errors;
0 points – the work is incomplete or contains significant errors.

Answer evaluation criteria:
2 points – the answer is complete, well-argued;
1 point – the answer is generally correct, but has flaws or minor errors;
0 points - there is no answer or the answer is incorrect.

Criteria for evaluating the timeliness of work submission for defense:
2 points – the work is presented 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 performing and defending computer practicals:
6 points × 8 comp. practice = 48 points.

The assignment for ***the modular test*** *consists of 3 questions - 1 theoretical and 2 practical. The answer to a theoretical question is worth 6 points, and the answer to a practical question is worth 10 points.*

Evaluation criteria for each theoretical test question:
6 points – the answer is correct, complete, well-argued;
5 points – the answer is correct, detailed, but not very well argued;
4 points - in general, the answer is correct, but has shortcomings;
3 points – there are minor errors in the answer;
1-2 points – there are significant errors in the answer;
0 points - there is no answer or the answer is incorrect.

Evaluation criteria for the practical test question:
9-10 points – the answer is correct, the calculations are completed in full;
7-8 points - the answer is correct, but not very well supported by calculations;
5-6 points - in general, the answer is correct, but has flaws;
3-4 points – there are minor errors in the answer;
1-2 points – there are significant errors in the answer;
0 points - there is no answer or the answer is incorrect.

The maximum number of points for a modular control work:
*2 papers * (6 points × 1 theoretical question + 10 points × 2 practical questions) = 52 points.*

The rating scale for the discipline is equal to:
$R_c = R_{com.practice} + R_{MKR} = 48$ *points + 52 points = 100 points.*

Calendar control: is carried out twice a semester as a monitoring of the current state of fulfillment of the syllabus requirements.

At the first certification (7th week), the student receives "passed" if his current rating is at least 50% of the maximum number of points (20 points) that the student can receive before the first certification.
At the second certification (13th week), the student receives "passed" if his current rating is at least 50% of the maximum number of points (35 points) that the student can receive before the second certification.

Semester control: ***assessment***

Conditions for admission to semester control:
With a semester rating (R_C) of at least 60 points and the enrollment of all computer practical work, the graduate student receives credit "automatically" according to the table (Table of correspondence of rating points to grades on the university scale). Otherwise, he has to complete the credit control work.

Completion and protection of a computer workshop is a necessary condition for admission to the performance of credit control work.

A student can try to improve his grade by writing a credit test, in which case his points obtained for the semester are canceled ("hard" grading system).

The composition and evaluation criteria of the assessment test:

The test task *consists of 4 questions - 2 theoretical and 2 practical. The answer to each theoretical and practical question is evaluated by 25 points.*

Evaluation criteria for each theoretical test question:
24-25 points – the answer is correct, complete, well-argued;
21-23 points – the answer is correct, detailed, but not very well argued;
17-20 points - in general, the answer is correct, but has flaws;
12-16 points – there are minor errors in the answer;
1-11 points – there are significant errors in the answer;
0 points - there is no answer or the answer is incorrect.

Evaluation criteria for the practical test question:
24-25 points – the answer is correct, the calculations are completed in full;
21-23 points - the answer is correct, but not very well supported by calculations;
17-20 points - in general, the answer is correct, but has flaws;
12-16 points – there are minor errors in the answer;
1-11 points – there are significant errors in the answer;
0 points - there is no answer or the answer is incorrect.

The maximum number of points for a modular control work:
25 points × 2 theoretical questions + 25 points × 2 practical questions = 100 points.

Table of correspondence of rating points to grades on the university scale :

Scores	Grade
100-95	Excellent
94-85	Very good
84-75	Good
74-65	Satisfactory
64-60	Sufficient
Less than 60	Fail
Admission conditions not met	Not Graded

8. Additional information on the discipline (educational component)

The list of questions submitted for semester control is given to students in the last lesson.

Work program of the academic discipline (syllabus):

Is designed by Ph.D., Assoc. Prof., Onai M.V.

Adopted by Computer Systems Software Department (protocol № 12, 11 February 2026)

Approved by the Methodical commission of the Faculty of Applied Mathematics (protocol № 7, 23 February 2026)