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

**Department of
System Programming and
Specialized Computer Systems**

Theory of Formal Languages and Parsing

Working program of educational discipline (syllabus)

Requisites of educational discipline

Higher education level	***First (bachelor)***
Knowledge domain	*12 Інформаційні технології*
Speciality	*123 Комп'ютерна інженерія*
Educational program	*System Programming and Specialized Computer Systems*
Status of the discipline	*Selective*
Form of education	*Full-time (day-time)*
Year of preparation, semester	*III year, spring semester*
Teaching hours	*4 credits ECTS, 120 hours*
Semester control / control activities	*Semester test, modular test, calculation-graphical work*
Schedule	*http://rozklad.kpi.ua*
Language of study	*English*
Information about supervisor of the course / professors	Lectures: Ph.D., *assoc.prof., Oleksandr I. Marchenko, pp.opt.2020@gmail.com, мел. +38(044)204-99-00* Lab classes: *assist.prof. Oleksii Marchenko, marchenko.oleksii@lll.kpi.ua.*
Discipline placement	*1. "Fundamentals of Translator Design: Lecture Notes" on the website of the Igor Sikorsky Kyiv Polytechnic Institute library* https://ela.kpi.ua/handle/123456789/45710 *2. Textbook "Theory of Formal Languages and Syntactic Analysis: Laboratory Works and Computational-Graphical Work" on the website of the Igor Sikorsky Kyiv Polytechnic Institute library* https://ela.kpi.ua/handle/123456789/54665 *3. Youtube channel «Олександр Іванович Марченко»* https://www.youtube.com/@Oleksandr_Marchenko *4. Distance learning course "Theory of Formal Languages and Syntactic Analysis" in the Moodle system:* https://scs-kpi.pp.ua/course/view.php?id=143 *5. Campus of the Igor Sikorsky KPI* https://campus.kpi.ua

Program of educational discipline

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

Studying «Theory of formal languages and parsing» subject allows to from for students competencies which are necessary for solving tasks of professional activity that concerns applying knowledge of formal languages and parsing theory for design of real translators.

***The purpose** of the discipline is to form the following students abilities:*

- *to analyze programming languages grammars from the formal point of view with the aim to determine the class a grammar belongs to;*
- *to choose the most applicable scanning and parsing methods for translator design for a given programming language;*
- *to choose grammar classes which are the most applicable for creation of a particular programming language.*

The subject *of the discipline is formal grammar classes, scanning and parsing methods and techniques which are directed to improvement of theoretical level of system programmers and their deep understanding theoretical basics of translator design.*

According to the requirements of the educational and professional program, students after mastering the discipline "Theory of formal languages and parsing" must demonstrate the following learning outcomes:

KNOWLEDGE of:

- *main types of translators and their structure;*
- *theory of formal grammars;*
- *correspondence of classes of formal grammars and parsing algorithms;*
- *correspondence of types of state machines to classes of formal grammars;*
- *methods and algorithms for scanners and parsers design;*
- *techniques of internal representation of programs in translators.*

SKILLS*:*

- *analyze grammars of programming languages from a formal point of view in order to determine the class to which this grammar belongs and, accordingly, the scanning and parsing methods;*
- *develop scanners;*
- *develop tables of different types of parsers.*

EXPERIENCE:

- *creation and adjustment new programming languages grammars;*
- *use of knowledge of the theory of formal languages and parsing in the practice of developing scanners and parsers of various types and classes.*

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

Prerequisites *- The discipline "Theory of formal languages and parsing" is based on the material of the disciplines "Mathematical analysis", "Applied theory of digital automata", "Data structures and algorithms", "Programming" and "System programming". A student must have knowledge and skills according to the educational plan of these disciplines.*

Post-requisites *- The discipline "Theory of formal languages and parsing" provides the study of the disciplines "Computer architecture" and "System software" of the educational plan of the educational degree "Bachelor" in the specialty 123 - "Computer engineering".*

3. The content of the discipline

CHAPTER 1. Basic concepts and implementation schemes of translators.

Topic 1.1. Translator, compiler, interpreter and assembler.

Topic 1.2. Generalized translator structure.

Topic 1.3. Concepts of lexical analysis, parsing, optimization and output code generation.

CHAPTER 2. Basic concepts of the theory of formal grammars.

Topic 2.1. Concept of formal grammar and basic terms.

Topic 2.2. Methods of writing formal grammars: BNF, MBNF, regular expressions, Wirth diagrams.

Topic 2.3. The concept of string generation, the concept of formal language, sentential form and sentence.

Topic 2.4. Chomsky's classification of formal grammars and languages.

Topic 2.5. Properties of context-free grammars. Chomsky's normal form. Greibach’s normal form.

Topic 2.6. Equivalent and non-equivalent grammars.

Topic 2.7. Unambiguous and ambiguous grammars.

CHAPTER 3. Formal grammars and lexical analysis.

Topic 3.1. Lexical analysis and automatic grammars (type 3).

Topic 3.2. Automaton (regular) grammars (type 3) and finite state automaton (machine).

Topic 3.3. Transition from a non-deterministic automaton to a deterministic one.

Topic 3.4. Regular expressions and transformation of regular expressions into finite state automata.

Topic 3.5. A lexical analyzer as an implementation of a finite state machine (automaton) corresponding to a type 3 grammar.

CHAPTER 4. Parsing and pushdown automata (PDA).

Topic 4.1. The concept of PDA and its structural diagram.

Topic 4.2. Configuration and operating cycle of PDA, initial and final configurations of PDA.

Topic 4.3. Algorithm of non-deterministic PDA.

Topic 4.4. Context-free grammars and PDA.

Topic 4.5. Deterministic PDA and deterministic context-free languages.

CHAPTER 5. Strategies and algorithms of syntactic analysis and parsing.

Topic 5.1. Problems of parsing. Left and right outputs.

Topic 5.2. Top-down and bottom-up parsing strategies. The concept of the direct reducible phrase and the base.

Topic 5.3. Definition of LL(k)-, LR(k)-, RR(k)- and RL(k)-grammars.

Topic 5.4. Algorithms of parsing according to the descending strategy.

Topic 5.4.1. The concept of FIRST and FOLLOW sets.

Topic 5.4.2. Conditions for parsing by the top-down strategy without returns.

Topic 5.4.3. Algorithm of parsing by the method of recursive descent.

Topic 5.4.4. Algorithm of parsing using the Knuth machine method.

Topic 5.4.5. Algorithm of parsing according to the tabular method of predictive parsing.

Topic 5.5. Algorithms of parsing using the bottom-up strategy.

Topic 5.5.1. Algorithm of parsing by the method of precedence grammars.

Topic 5.5.2. Algorithm of parsing using the shift-reduce method.

Topic 5.6. Mixed parsing algorithms. Modified Irons' algorithm.

SECTION 6. Methods of internal representation of a source program.

Topic 6.1. Parsing tree. Linear forms of representation of a parsing tree.

Topic 6.2. Semantic tree, attribute tree.

Topic 6.3. Triads, tetrads, Reverse Polish notation (RPN).

SECTION 7. General concept of semantic analysis and code generation.

Topic 7.1. Formal semantics of programming languages. Ways of describing semantics.

Topic 7.2. Description of semantics on the parse tree. Semantic definitions. A simple metasemantic language.

Topic 7.3. Description of semantics by the method of recursive semantic procedures.

Topic 7.4. Semantic definitions of statements and data structures of high-level programming languages.

SECTION 8. Translator design systems (compiler compilers).

8.1. Compiler design system CBS (Compiler Building System).

8.2. Compiler design systems YACC, BISON, ZUBR.

4. Training materials and resources

Basic literature

1. V.J. Rayward-Smith. A first Course in Formal Language Theory. Blackwell Scientific Publications, 1983.– 124 p.

2. Alfred V. Aho, Jeffrey D. Ullman. The Theory of Parsing, Translation and Compiling. Volume 1. Parsing. Prentice-Hall, Inc., Englewood Cliffs, N.J., 1972. – 582 p.

3. Alfred V. Aho, Jeffrey D. Ullman. The Theory of Parsing, Translation and Compiling. Volume 2. Compiling. Prentice-Hall, Inc., Englewood Cliffs, N.J., 1973. – 471 p.3. Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman. Compilers: Principles, Techniques, & Tools. Pearson Education, Inc., 2007. – 1009 p.

4. Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman. Compilers: Principles, Techniques, & Tools. Pearson Education, Inc., 2007. – 1009 p.

5. Основи проектування трансляторів: Конспект лекцій : [Електронний ресурс] : навч. посіб. для студ. спеціальності 123 – «Комп'ютерна інженерія» / О. І. Марченко ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 2,64 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2021. – 108 с. – Режим доступу до ресурсу: https://ela.kpi.ua/handle/123456789/45710

6. Теорія формальних мов та синтаксичного аналізу: Лабораторний практикум і розрахунково-графічна робота з дисципліни «Теорія формальних мов та синтаксичного аналізу» : [Електронний ресурс] : навч. посіб. для студ. спеціальності 123 – «Комп'ютерна інженерія» / О. І. Марченко, О. О. Марченко ; КПІ ім. Ігоря Сікорсь-кого. – Електронні текстові дані (1 файл: 1,12 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2023. – 121 с. – Режим доступу до ресурсу: https://ela.kpi.ua/handle/123456789/54665

Additional literature (elective / familiarization)

1. P.M. Lewis, D.J. Rosenkrantz, R.E. Stearns, Compiler Design Theory. Addison-Wesley Publishing Company, Inc., 1976. – 645 p.

2. W.M. McKeeman, J.J. Horning, D.B. Wortman. A Compiler Generator. Prentice-Hall, Inc., 1979. – 513 p.

3. David Gries. Compiler Construction for Digital Computers. John Wiley & Sons, Inc., 1971. – 542 p.

4. Donald Knuth. Top-Down Syntax Analysis, Acta Informatica, volume 1, 1971. – pp. 79-110.

5. Leland L. Beck. System Software. An Introduction to systems programming. Addison-Wesley Publishing Company, Inc., 1985. – 437 p.

Information resources

1. Youtube channel «Олександр Іванович Марченко»
https://www.youtube.com/@Oleksandr_Marchenko

2. Distance learning course "Theory of Formal Languages and Syntactic Analysis" in the Moodle system: https://scs-kpi.pp.ua/course/view.php?id=143

3. Electronic campus of the NTUU "Igor Sikorsky KPI". Матеріали з дисципліни «Теорія формальних мов та синтаксичного аналізу». – Access link : http://login.kpi.ua.

Educational content

5. Methods of mastering the discipline (educational component)

Lectures

№ з/п	*Chapters and Topics*
1	*Introduction.* *CHAPTER 1. Basic concepts and implementation schemes of translators.* *Topic 1.1. Translator, compiler, interpreter and assembler.* *Topic 1.2. Generalized translator structure.* *Topic 1.3. Concepts of lexical analysis, parsing, optimization and output code generation.*
2	*CHAPTER 2. Basic concepts of the theory of formal grammars.* *Topic 2.1. Concept of formal grammar and basic terms.* *Topic 2.2. Methods of writing formal grammars: BNF, MBNF, regular expressions, Wirth diagrams.* *Topic 2.3. The concept of string generation, the concept of formal language, sentential form and sentence.*
3	*Topic 2.4. Chomsky's classification of formal grammars and languages.* *Topic 2.5. Properties of context-free grammars. Chomsky's normal form. Greibach's normal form.* *Topic 2.6. Equivalent and non-equivalent grammars.* *Topic 2.7. Unambiguous and ambiguous grammars.*
4	*CHAPTER 3. Formal grammars and lexical analysis.* *Topic 3.1. Lexical analysis and automatic grammars (type 3).* *Topic 3.2. Automaton (regular) grammars (type 3) and finite state automaton (machine).*
5	*Topic 3.3. Transition from a non-deterministic automaton to a deterministic one.* *Topic 3.4. Regular expressions and transformation of regular expressions into finite state automata.*
6	*Topic 3.5. A lexical analyzer as an implementation of a finite state machine (automaton) corresponding to a type 3 grammar.*
7	*CHAPTER 4. Parsing and pushdown automata (PDA).* *Topic 4.1. The concept of PDA and its structural diagram.* *Topic 4.2. Configuration and operating cycle of PDA, initial and final configurations of PDA.* *Topic 4.3. Algorithm of non-deterministic PDA.*
8	*Topic 4.4. Context-free grammars and PDA.* *Topic 4.5. Deterministic PDA and deterministic context-free languages.*
9	*CHAPTER 5. Strategies and algorithms of syntactic analysis and parsing.* *Topic 5.1. Problems of parsing. Left and right outputs.* *Topic 5.2. Top-down and bottom-up parsing strategies. The concept of the direct reducible phrase and the base.* *Topic 5.3. Definition of LL(k)-, LR(k)-, RR(k)- and RL(k)-grammars.*
10	*Topic 5.4. Algorithms of parsing according to the descending strategy.* *Topic 5.4.1. The concept of FIRST and FOLLOW sets.* *Topic 5.4.2. Conditions for parsing by the top-down strategy without returns.* *Topic 5.4.3. Algorithm of parsing by the method of recursive descent.*
11	*Topic 5.4.4. Algorithm of parsing using the Knuth machine method.* *Topic 5.4.5. Algorithm of parsing according to the tabular method of predictive parsing.*
12	*Topic 5.5. Algorithms of parsing using the bottom-up strategy.* *Topic 5.5.1. Algorithm of parsing by the method of precedence grammars.*
13	*Topic 5.5.2. Algorithm of parsing using the shift-reduce method.* *Topic 5.6. Mixed parsing algorithms. Modified Irons' algorithm.*

14	*SECTION 6. Methods of internal representation of a source program.* *Topic 6.1. Parsing tree. Linear forms of representation of a parsing tree.* *Topic 6.2. Semantic tree, attribute tree.* *Topic 6.3. Triads, tetrads, Reverse Polish notation (RPN).*
15	*SECTION 7. General concept of semantic analysis and code generation.* *Topic 7.1. Formal semantics of programming languages. Ways of describing semantics.* *Topic 7.2. Description of semantics on the parse tree. Semantic definitions. A simple metasemantic language.*
16	*Topic 7.3. Description of semantics by the method of recursive semantic procedures.*
17	*Topic 7.4. Semantic definitions of statements and data structures of high-level programming languages.*
18	*SECTION 8. Translator design systems (compiler compilers).* *8.1. Compiler design system CBS (Compiler Building System).* *8.2. Compiler design systems YACC, BISON, ZUBR.*

Laboratory works (computer practicum)

The main purposes of the set of laboratory works: gaining experience and skills of designing of translator parts by students.

№	*Name of laboratory work (computer practicum)*	*Teaching hours*
1	*Laboratory work. Scanner development.*	*6*
2	*Laboratory work. Parser development by the recursive descent method according to top-down strategy.*	*6*
3	*Laboratory work. Parser development by the Knuth analytical machine method according to top-down strategy.*	*6*

6. Self-students studying (SSS)

№	*The name of the topic submitted for independent study*	*Teaching hours of SSS*
1	*Scanner development.*	*16*
2	*Parser development by the recursive descent method according to top-down strategy.*	*16*
3	*Parser development by the Knuth analytical machine method according to top-down strategy.*	*16*
4	*Modified Irons' algorithm.*	*4*
5	*Semantic and attribute trees.*	*4*
6	*Generation of the output code for the main control structures of algorithms.*	*10*

Policy and control

7. Policy of academic discipline (educational component

Attending classes. *All students must attend lectures and laboratory classes, where they must actively work on learning the studied material. For objective reasons (for example, quarantine, martial law), educational process can take place online. At the same time, absence from a lecture, practical, or laboratory session does not result in penalty points, since the student's final rating score is formed solely on the basis of the assessment of learning outcomes. Performance of control tasks in a practical or laboratory tasks is evaluated and must be completed at another time, if the absence was due to a valid reason.*

Missed evaluation control measures. *Each student has the right to work out missed for a good reason (sick leave, mobility, etc.) classes through independent work. Details on the link: https://kpi.ua/files/n3277.pdf.*

Procedure for appealing the results of evaluation control measures. *The student can raise any issue related to the control procedure and expect it to be considered according to predefined procedures. Students have the right to challenge the results of control measures, explaining which criterion they do not agree with according to the assessment.*

Calendar control *is carried out in order to improve the quality of students' education and monitor the student's fulfillment of the syllabus requirements.*

All individual laboratory works *must be performed independently and reports must be submitted in compliance with the established deadlines for each laboratory work. The practical results of a laboratory work must be confirmed by knowledge of the theoretical material and practical skills on the topic during the defense. Detailed requirements are given in the next section.*

Deadlines and re-pass. Works that are submitted in violation of the terms (deadlines) without valid reasons are evaluated at a lower grade. The re-pass of the modular control works take place with the permission of the dean's office if there are valid reasons (for example, sick leave). Without valid reasons, a missed test is assessed as zero points.

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

Norms of ethical behaviour. *Norms of ethical behaviour of students and employees are defined in Section 2 of the Code of Honour of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute". Details: https://kpi.ua/code.*

Inclusive education. *The acquisition of knowledge and skills during the study of the discipline "Foundations of Sustainable Development" may be available to most people with special educational needs, except for students with severe visual impairments who do not allow to perform tasks with personal computers, laptops, and/or other technical means.*

Learning a foreign language. *During the assignments, students may be encouraged to refer to Ukrainian-language sources.*

Assignment of incentive points. *According to the Regulations on the system of assessment of learning outcomes, the sum of all incentive points may not exceed 10% of the rating scale.*

8. Types of control and rating system for assessing learning outcomes (RSA)

The student's rating in the discipline consists of:

1) points for performing and defending 3 laboratory works;

2) points for passing one modular control work;

SYSTEM OF RATING POINTS

1. Laboratory works

During the semester, students perform 3 laboratory works.

The maximum number of points for laboratory works 1 and 2: 25 points.

Points are awarded for:

- quality of the report design: 0-2 points;

- quality of program execution: 0-10 points;

- answer during the defense of laboratory work: 0-8 points;

- timely presentation of work for defense: 0-5 points.

The maximum number of points for laboratory work 3: 30 points.

Points are awarded for:

28-30 points - the solution is correct;

20-27 points – there are minor errors in the decision, but the course of the decision is correct;

5-19 points – there are gross errors in the decision;

0-4 points – there is no solution or the solution is incorrect.

The maximum number of points for all laboratory works:

*25 points * 2 lab works + 30 points * 1 lab work =* ***80 points****.*

Procedure for defending laboratory work:

1. The work is considered ready for defense if the program is written according to its version, its code and report are uploaded to the Moodle distance learning system of the SP&SCS Department, the code is successfully compiled, the report meets the requirements (contains all information).

2. The order of defending laboratory work in a class is established one by one for each laboratory work, that is, the same student can defend another laboratory work only in the absence of other students who wish to defend the first laboratory work in this class.

3. If a student missed the deadline for defending a certain laboratory work, then in the following classes he/she will defend it after the students who have already prepared the next laboratory work on time and defend it before the deadline for this laboratory work.

4. If during the defense of the work, it is revealed that it was not done independently, the laboratory work is not credited and the student receives 0 (zero) points for it.

2. Modular control

During the semester, students perform one modular control work (MCW).

The modular control work can be carried out either in writing in the classroom during intramural form of education, or in writing with sending the work by mail by the specified time, or in the form of tests through the Moodle distance learning system of the SP&SCS Department.

The maximum number of points for a modular control work:

- ***20 points*** *if the student performs the work during the first MCR;*

- ***17 points*** *if the student performs the work during the second MCR;*

- ***14 points*** *if the student performs the work during the third MCR in the credit week before the session;*

- ***10 points*** *if the student performs the MCR in an additional session.*

A student has only one attempt during only one of the periods of writing an assignment or opening tests in the Moodle system, i.e. assignments are NOT rewritten and tests are NOT retaken.

Evaluation criteria:

0-5 points *- the task is completed or not completed in the allotted time (minus point for each minute of delay);*

If the work is received from the student later than 5 minutes after the established deadline, then this work is not counted at all.

0-15 points *- correctness and completeness of the task:*

- *14-15 points - the solution is correct or practically correct;*
- *11-13 points - there are minor errors in the solution, but the course of the solution is correct;*
- *8-10 points - there are gross errors in the solution;*
- *2-7 points - only some elements of the solution are correct;*
- *0-1 point - there is no solution or the solution is incorrect.*

In subsequent MCRs, the scores change in proportion to the corresponding maximum score.

3. Calculation of the rating scale (R).

The maximum rating ***R = 100 points*** *and is determined as the sum of positive points received for completing and defending laboratory work and writing a module test.*

4. Calendar control

At the first calendar control, the student receives "passed" if his current rating is at least 22 points (50% of the maximum number of points a student can receive before the first certification).

At the second calendar control, the student receives "passed" if his current rating is at least 37 points (50% of the maximum number of points a student can receive before the second certification).

5. Conditions for admission to the semestral test and determining the final mark

The sum of the weighted points of control measures during the semester is:

R = 50 + 30 + 20 = 100 points.

Ehe total rating cannot exceed 100 points.

A necessary condition for admission to the semestral test is the enrollment of two first laboratory work and calculation-graphical work, as well as the starting rating of RD at least 40% of R, i.e. 40 points.

If the student received a rating of at least 0.6 R, he has the right to receive semestral test "automatically" or to write a test paper at the level of others in order to increase the rating.

In the case of writing a test paper, points for the test paper are added to the points for laboratory and calculation-graphical work (without points for MCW) and this rating is final.

Maximum number of points for test paper: 20 points.

The semestral test ticket consists of 3 tasks. The first two tasks are valued at 7 points, the third task is valued at 6 points.

Evaluation criteria for the first two tasks of credit work:

7 – correct and meaningful answer;

5-6 – the answer is meaningful, but has minor flaws;

2-4 – the answer is incomplete or contains errors;

0-1 – no answer or the answer is incorrect.

Evaluation criteria for the first two tasks of credit work:

6 – correct and meaningful answer;

4-5 – the answer is meaningful, but has minor flaws;

2-3 – the answer is incomplete or contains errors;

0-1 – no answer or the answer is incorrect.

Table of correspondence of rating points to final marks according to the university scale:

Points	*Final mark*
100-95	Excellent
94-85	Very good
84-75	Good
74-65	Satisfactory
64-60	Enough
Less than 60	Unsatisfactory
Prerequisite for admission to the exam is not completed	Not admitted

Working program of educational discipline (syllabus):

Developed by:

assoc. prof. of the Department of System Programming and Specialized Computer Systems, Ph.D., assoc. prof., Oleksandr I. Marchenko, and

assist. prof. of the Department of System Programming and Specialized Computer Systems, Oleksii O. Marchenko

Approved by the Department of System Programming and Specialized Computer Systems (protocol № 9 of 27.01.2026)

Agreed by Methodical Council of the Faculty of Software Systems and Applied Mathematics (protocol №7 of 23.02.2026)