



Engineering Calculation Software
Work program of the educational component (Syllabus)

Details of the educational component

Higher education level	<i>First (educational and professional)</i>
Field of Knowledge	<i>13 Mechanical Engineering</i>
Speciality	<i>133 Branch Mechanical Engineering</i>
Educational program	<i>Engineering of equipment for the production of polymer and building materials and products</i>
Status of the educational component	<i>Mandatory</i>
Form of study	<i>full-time (full-time)</i>
Year of preparation, semester	<i>2nd year, 3rd semester</i>
Scope of the educational component	<i>7kr. / 210 hrs. (90 auditorium, 120 SRS)</i>
Semester control / control measures	<i>Exam</i>
Class schedule	<i>5 hours per week (2 hours of lectures and 3 hours of laboratory classes)</i>
Language of instruction	<i>Ukrainian</i>
Information about the course leader / teachers	<i>http://intellect.cpsm.kpi.ua/profile/sde9</i>
Course placement	<i>Certified distance learning course on the Sikorsky platform</i>

Program of the educational component

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

The student receives basic knowledge of computer science, engineering calculations on computers, rules for working with a computer, the basics of programming; will be able to perform calculations and evaluate the results obtained in software environments, MSDev FPS, Fortran, or similar.

The subject is engineering calculations using a programming language specialized for scientific and engineering calculations.

The purpose of the educational component is to participate in the formation of the following competencies in accordance with the requirements of the educational program.

Integral competence:

- The ability of a person to solve complex specialized tasks and practical problems in a certain field of professional activity or in the process of training, which involves the application of certain theories and methods of the relevant sciences and is characterized by complexity and uncertainty of conditions.*

General competencies:

Ability to think abstractly;

Skills in the use of information and communication technologies.

Professional competencies:

- Ability to apply typical analytical methods and computer software tools for solving engineering problems of industrial engineering, effective quantitative methods of mathematics, physics, engineering sciences, as well as appropriate computer software for solving engineering problems of industrial engineering;*

- Ability to perform engineering calculations on a computer using a high-level algorithmic language (Fortran, etc.).

1.2. According to the requirements of the educational program, students, after mastering the educational component "Software of Engineering Calculations", must demonstrate the following program learning outcomes:

- Carry out engineering calculations to solve complex problems and practical problems in industrial engineering;
- Read, analyze, edit source program code, compile programs for engineering calculations on computers using a high-level algorithmic language (Fortran, etc.) and appropriate compilers, receive from the source program code an object file and an executable file and the results of its work.

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

The educational component "Software of Engineering Calculations" is preceded by "Informatics". The educational component "Software of Engineering Calculations" provides educational components "Applied Mechanics of a Solid Deformed Body", "Diploma Design".

3. Content of the educational component

Introductory provisions. Rating system. Safety and ergonomics when working on a computer

Chapter 1. Operating system and the basics of computer functioning

Topic 1.1. Technical means of computers. Operating system. Team. File management.

Chapter 2. Fundamentals of programming in the algorithmic language Fortran

Topic 2.1. General information about the algorithmic language Fortran. Compiler file structure. Compile source code.

Topic 2.2. Algorithm. Types of algorithms. Drawing up flowcharts. Table of identifiers.

Topic 2.3. Algorithmic language Fortran. Programming Basics.

Chapter 3. Working with spreadsheets. IDE. MS Dev, FPS programming system. Procedural programming

Topic 3.1 Working with spreadsheets.

Topic 3.2. IDE. MS Dev environment and modern Fortran in FPS 4 implementation.

Topic 3.3. Program units

Topic 3.4. Data transfer between software units

Chapter 4. Modeling Basics

Topic 4.1. Modeling method

Topic 4.2. Models based on SDR and methods for their solution

4. Training materials and resources

Basic literature

1. Software of engineering calculations. Laboratory practicum [Electronic resource] : a textbook for the training of bachelors who study in the specialty 133 – "Industrial Engineering", according to the educational program "Computer-integrated technologies for designing chemical engineering equipment" / Igor Sikorsky Kyiv Polytechnic Institute; compiled by D. E. Sidorov, I. O. Kazak. – Electronic text data (1 file: 3.24 MB). – Kyiv: Igor Sikorsky Kyiv Polytechnic Institute, 2022. – 166 p. Access: <https://ela.kpi.ua/handle/123456789/50647>

2. Software of engineering calculations. Lectures [Electronic resource] : a textbook for the preparation of bachelors who study in the specialty 133 – "Industrial Engineering", educational program "Computer-integrated technologies for designing chemical engineering equipment" / Igor Sikorsky Kyiv Polytechnic Institute; compiled by D. E. Sidorov, I. O. Kazak. – Electronic text data (1 file: 3.78 M bytes). – Kyiv: Igor Sikorsky Kyiv Polytechnic Institute, 2022. – 244 p. Access: <https://ela.kpi.ua/handle/123456789/50645>

3. *Engineering calculations on computers: Tasks for independent solution. Teaching aids for the training of bachelors in the specialties 131 – Applied Mechanics, 133 – Branch Mechanical Engineering / Compiled by: D.E. Sidorov, I.O. Kazak. – Kyiv: I. Sikorsky Kyiv Polytechnic Institute, 2019. – 99 p. Access: <http://ela.kpi.ua/handle/123456789/27600>*

4. *Engineering calculations on computers. Part 1. Programming. Laboratory workshop on the educational component: teaching aids for the training of full-time bachelors in the specialty 133 – Industrial Engineering / Compiled by D.E. Sidorov, I.O. Kazak, I.I. Ivitsky. – Kyiv: I. Sikorsky Kyiv Polytechnic Institute, 2018. – 66 p. Access: <http://ela.kpi.ua/handle/123456789/25145>*

5. *Engineering Calculations on Computers. Part 2. Procedural Programming. Laboratory Practicum on the Educational Component: Teaching Aids for the Training of Full-time Bachelors in the Specialty 133 – Branch Mechanical Engineering of Mechanical Engineering / Compiled by D.E. Sidorov, I.O. Kazak, I.I. Ivitsky. – Kyiv: I. Sikorsky Kyiv Polytechnic Institute, 2018. – 38 p. Access: <http://ela.kpi.ua/handle/123456789/25146>*

6. D.E. Sidorov, and I.O. Kazak. *Basics of engineering calculations on computers. Programming in the algorithmic language FORTRAN. – Kyiv: Center for Educational Literature, 2016. – 185 p. (in Russian).*

Further reading

1. *Methodical instructions for performing tasks on laboratory work and independent work of students in the credit module "Engineering calculations on computers – 1. Programming" / compiled by: D.E. Sidorov, I.O. Kazak. – Kyiv: NTUU "KPI them. Igor Sikorsky", 2016. – 86 p. (in Russian).*

2. *Methodical instructions for performing tasks on laboratory work and independent work of students on the credit module "Engineering calculations on a computer - 2. Procedural programming and the basics of modeling" / Compiled by: D.E. Sidorov, I.O. Kazak. – Kyiv: NTUU "KPI them. Igor Sikorsky", 2016. – 48 p. (in Russian).*

3. *Informatics: Excel and Basic for Application: Manual. Manual. / Lopotko O.V. Recommended by the Ministry of Education and Science of Ukraine. Kyiv: Karavela Publ., 2018. 272 c.*

4. *Chapman, Stephen J. Fortran for scientists and engineers / Stephen J. Chapman, BAE Systems Australia. Fourth edition. | New York, NY : McGraw-Hill, a business unit of The McGraw-Hill Companies, Inc. (March 16, 2017) ISBN 978-0-07-338589-1; MHID 0-07-338589-1*

5. *Methodical instructions for the implementation of tasks from the computer workshop and independent work of students in the discipline "Engineering calculations on a computer – 1. Programming", for students of the areas of training 6.050503 "Mechanical Engineering", 6.050502 "Engineering Mechanics" / Compiled by: D.E. Sidorov, I.O. Kazak. – KYIV: NTUU "KPI", 2012. – 86 p. (in Russian).*

6. *Engineering calculations on personal electronic computers – 2. Procedural Programming and the Basics of Modeling: Methodical Instructions for Performing the Tasks of Computer Practicum and Independent Work of Students in the Areas of Training 6.050503 – Mechanical Engineering, 6.050502 – Engineering Mechanics / Compiled by: D.E. Sidorov, I.O. Kazak. – KYIV: NTUU "KPI", 2013. – 48 p. (in Russian).*

Information resources on the Internet

1. *Work program of the educational component (syllabus) - in the "Campus" system, on the "Sikorsky" platform and on the website of the department;*

2. *Rating system of assessment in the "Campus" system on the "Sikorsky" platform;*

3. *Synopsis of lectures on the Sikorsky platform;*

4. *Methodological guidelines and teaching aids:*

<http://ela.kpi.ua/handle/123456789/27600>

<http://ela.kpi.ua/handle/123456789/25145>

<http://ela.kpi.ua/handle/123456789/25146>

<https://ela.kpi.ua/handle/123456789/50647>

<https://ela.kpi.ua/handle/123456789/50645>

<https://fortranwiki.org/fortran/show/Fortran+Courses>

<http://faculty.washington.edu/rjl/uwhpsc-coursera/fortran.html>

5. Distance courses:

<https://www.udemy.com/course/fortran-for-beginners/>
<https://siit.co/courses/fortran-course-and-certification/414>
<https://www.udemy.com/course/intermediate-fortran/>
<https://www.udemy.com/course/fortran-course/>
<https://www.classcentral.com/subject/fortran?certificate=true>
<https://www.classcentral.com/course/fortran-for-scientific-computing-56029>
<https://www.futurelearn.com/courses/fortran-for-scientific-computing>

Educational content

5. Methods of mastering the educational component

Lectures (36 hours)

Lectures are aimed at:

- providing modern, holistic, interdependent knowledge on the educational component "Software of Engineering Calculations", the level of which is determined by the target setting for each specific topic;
- ensuring the creative work of bachelors together with the teacher during the lecture;
- education of bachelors' professional and business qualities and the development of independent creative thinking in them;
- formation of the necessary interest in bachelors and providing direction for independent work;
- reflection of methodological processing of the material (highlighting the main provisions, conclusions, recommendations, clear and adequate to their formulation);
- use of visual materials for demonstration, combining, if possible, them with the demonstration of results and samples;
- teaching the materials of the discipline in a clear and high-quality language in compliance with structural and logical connections, explaining all newly introduced terms and concepts;
- accessibility for perception by this audience.

No. s/p	Title of the lecture topic and list of main questions	Hours
1.	Introductory provisions. Rating system. Safety and ergonomics when working on a computer Objectives of the course, the role and place of discipline in the structure of the educational process. The use of computers in solving mathematical models. Modeling and engineering. Achievements in the field of calculations and modeling of technical objects. Literature recommended for study. Provisions of the rating system for assessing student performance in the implementation of the module program.	2
2.	Technical means of computers and OS Technical means of computers. BIOS. Self-diagnosis of the computer when turned on. OS. OS functions. OS classification, examples. OS program structure (examples of the simplest single-tasking MS-DOS and WIN).	2
3.	Working in the console File types. Console. OS commands. Automate work with batch command files. File management.	2
4.	Classification of translators and the city of Fortran in her Interpreters and compilers. High and low level languages. Algorithmic language Fortran, use in technical calculations, roots and prospects for development. The composition of the Fortran programming system with examples. Broadcast procedure. Obtaining *.obj and *.exe modules. Compile batch command files to control the compilation process.	2

5.	Algorithmic language Fortran. Rules for generating a source code file Algorithm, requirements for the algorithm. Flowchart of the representation of the algorithm Table of identifiers. Alphabet and syntax. Priorities of arithmetic operations. Fortran screen. Rules for writing operators. Executable and non-executable statements. Label, comments, hyphenation. Output in a simple form. The structure of the program, the simplest program. Operators PROGRAM, STOP, END. I/O in a simple form. I/O operators. Dialog mode of the program.	2
6.	Data type description operators Data types. Operators INTEGER, REAL, COMPLEX, CHARACTER, LOGICAL. Rules for working with data of the same type and different types. Assignment and default rules for data types. IMPLICIT operator. Giving a name to data: constants and variables, PARAMETER, length name, assignment rules. Features that are standard. Examples.	2
7.	Conditional control operators. Branching Control operators. Unconditional transition. Three types of conditional control operators. Arithmetic and logical conditional control operators. Boolean expression. Relationship. Logical actions.	2
8.	Block conditional control operator. Block conditional control operator. Determination of the logical level of the operator. Multivariate branching.	2
9.	Cyclic calculations Cycles. Organization of cyclical processes. Arrays. Static and dynamic data.	2
10.	Formatted I/O* Formatted I/O. Format specifiers for real and integer. Format specifiers for non-numeric data.	(2*)
11.	Working with the computer code table* Computer code table, pseudographics. Formation of a tabular output. Metacommands. Program code optimization. Search and work out errors of the "run-time error" type.	(2*)
12.	Working with spreadsheets Working with spreadsheets and building diagrams. Import and export data using text files.	2
13.	IDE on the example of MS Dev and modern Fortran in the implementation of FPS 4 Purpose of IDE, examples of IDEs, MS Dev. FPS capabilities 4. Working with visual tools for software development MS Dev. Project. Project files. BUILD menu. Compiling project files. DEBUG and RELEASE compilation modes. Icon editor.	2
14.	Program units Ideology of procedural programming. Program units. Broadcasting of program units.	2
15.	Program units Main program. Subroutine. Formal and actual parameters of the Function. Types of functions. Block date..	2
16.	Data transfer between software units Object visibility zone. Transfer data between software units through parameter lists. Internal procedures. Transfer data between software units using shared memory units.	2
17.	Modules Application of modules. Expanding the visibility zone of objects.	2
18.	Working with files Transfer data between software units using a file interface.	2
19.	Modeling method Model definition and modeling. Generalized diagram of the modeling process. Object, physical model, mathematical model, solution. Accuracy of solution for mathematical models.	2
20.	Models based on SDR and methods for their solution Structure of analytical and numerical solution of mathematical models. Resistance.	2

	<i>Consistency. Convergence. Advantages of methods, disadvantages and features. Example.</i>	
21.	Technical models* <i>Modeling the process of heating (cooling) the body. Description of the real process, formulation of a physical model based on hypotheses, mathematical model. Euler's method for solving ADR. Application of the method for numerical solution of a composite mathematical model.</i>	(2*)
22.	Chemical reaction models * <i>Generalized equation of a chemical reaction. The law of acting masses is the basis for building a mathematical model. Differential model. Runge-Kutta method for numerical solution of ZDR. Application of a method to solve a composite model.</i>	(2*)
23.	Differential models in ecology * <i>Objects of research in ecological models. Physical model and its clarification, mathematical model, obtaining a solution and its analysis. (The method of Euler, Runge-Kutt or M. K. R. for solving a composite model.)</i> Economic models * <i>Advertising effectiveness. Logistics curve. Exact and numerical solution. Supply and demand. Objects of study, physical models, mathematical models, solutions and analysis.</i>	(2*)
	Just	36

*-Optional

Computer Workshop (54 hours)

The purpose of the computer workshop on the educational component is to consolidate theoretical knowledge, form skills and abilities of engineering calculations, programming, solve computational problems, and present the results of calculations.

No. s/p	Job Title	Points	Number of auditoriums. Hours
1.	<i>Introduction. Safety precautions. Introductory provisions.</i>	2	2
2.	<i>Technical means of computers. Operating system. Some operating system commands. File management.</i>	4	4
3.	<i>Work with the Fortran programming system.</i>	2	4
4.	<i>Input and output operations, calculation of arithmetic expressions</i>	4	4
5.	<i>Conditional control operators, unconditional transition operator</i>	4	6
6.	<i>Cyclic processes. Organizing loops using IF and DO statements</i>	4	4
7.	<i>Format output. Building tables. Working with the table of computer codes. Working with arrays*.</i>	*	4*
8.	<i>Working with spreadsheets. Diagramming</i>	2	2
9.	<i>Work with the MSDev programming system (FPS 4).</i>	2	3
10.	<i>Icon Editor. Development and application of icons</i>	2	3
11.	<i>Design of the project structure using the procedural paradigm</i>	2	4
12.	<i>Creating programs using subroutines and functions:</i> <i>- data transfer using parameter lists;</i> <i>- data transfer using shared memory units;</i> <i>- the use of modules;</i> <i>- file interface.</i>	8	10
13.	<i>Solution of differential models based on ZDR according to the Euler (Runge-Kutt) method</i>	4	4
14.	<i>Modular control work</i>	10	2
15.	<i>Saturn Software.*</i>		2*

Just	50	54
------	----	----

*-Optional

6. Independent work of a student

Independent work takes 57% of the time spent studying the educational component, including preparation for the exam. The main task of independent work of bachelors is to master knowledge that is not included in the list of lecture questions through a personal search for information, the formation of an active interest in a creative approach in educational work. In the process of independent work within the framework of the educational component, the bachelor must learn to analyze modern sources, educational literature, and use certificate distance courses.

No. s/p	Name of the topic submitted to the SRC	Number of CPC hours
1.	Topic 1.1. Computer hardware. Operating system. Commands. File management. Preparation for laboratory work. Work in the locale and on the server. Study of the OS console and the features of working in console mode.	4
2.	Topic 2.1. General information about the algorithmic language Fortran. Compiler file structure. Compilation of source code. Preparation for laboratory work. Librarian. Step-by-step compilation by console commands. Alternative libraries.	4
3.	Topic 2.2. Algorithm. Types of algorithms. Drawing up flowcharts. Table of identifiers. Execution of engineering documentation when performing calculations on a computer. Images of blocks, alternative blocks, standard capabilities of text editors for the image of blocks of flowcharts.	4
4.	Topic 2.3. Algorithmic language Fortran. Fundamentals of programming Preparing for laboratory work. Data input and output operators. Double precision for standard data types. Changing the default rule. Programming console data entry. Redirecting program results. Boolean expressions. Boolean relations and logical operations. Loops with post- and prerequisites. Standard functions for working with arrays. Tasks and changing boundaries. Section of an array. Format specifiers for non-numeric data. Code tables of Cyrillic characters. Search and work out errors of the "run-time error" type.	30
5.	Topic 3.1 Working with spreadsheets. Excel functions. Visualization of the results of Fortran program calculations. Preparation for laboratory work. Software development for individual tasks.	4
6.	Topic 3.2. MS Dev environment and modern Fortran in FPS 4 implementation Features of Fortran 90. Preparation for computer workshops. Software development for individual tasks. FPS 4 format editor. MS Dev work with deleted files. Debugging program code.	10
7.	Topic 3.3. Program units Software development for individual tasks.	10
8.	Topic 3.4. Data transfer between software units Ways to place program units in project files. Data subroutines. Preparation for computer workshops. Software development for individual practical tasks.	10
9.	Topic 4.1. Modeling method Reduction in the transition from the original to the physical model with	2

	<i>examples.</i>	
10.	Topic 4.2. Models based on SDR and methods for solving them <i>Drawing up a mathematical model according to the task. Preparation and refinement of program code. Modification of the Euler method.</i>	6
11.	Preparation for the modular test work	4
12.	Exam preparation	30
	Just	120

You can independently complete the program of distance courses (Information resources on the Internet).

Policy and control

7. Educational Component Policy

Rules for attending classes and behavior in the classroom

Attendance at classes is a mandatory component of assessment. Bachelors are obliged to take an active part in the educational process, not to be late for classes and not to miss them without a good reason, not to interfere with the teacher's conduct, not to be distracted by actions not related to the educational process. The grade for the exam cannot be obtained (put on the list) without the student personally attending the exam according to the schedule.

Rules for assigning incentive and penalty points

Incentive points can be awarded by the teacher only for the performance of creative (optional) work in the discipline.

Additional points (up to 50 points, determined individually) or a course of laboratory work can also be credited by taking online specialized courses with the receipt of an appropriate certificate:

- <https://www.udemy.com/course/fortran-for-beginners/>
- <https://siit.co/courses/fortran-course-and-certification/414>
- <https://www.udemy.com/course/intermediate-fortran/>
- <https://www.udemy.com/course/fortran-course/>
- <https://www.classcentral.com/subject/fortran?certificate=true>
- <https://www.classcentral.com/course/fortran-for-scientific-computing-56029>
- <https://www.futurelearn.com/courses/fortran-for-scientific-computing>
- *Other courses can be enrolled in agreement with the teacher.*

Penalty points can be received for late completion or defense of laboratory work.

Deadline and retake policy

In case of debts from the educational component or any force majeure circumstances, bachelors must contact the teacher through the available (provided by the teacher) communication channels to solve problematic issues and agree on an algorithm of actions for working out.

Academic Integrity Policy

Plagiarism and other forms of dishonest work are unacceptable. Plagiarism includes the absence of references when using printed and electronic materials, quotes, opinions of other authors. All tasks designed for individual performance must be performed personally by the student. Hints and cheating during writing tests, control, conducting classes are unacceptable; defense of laboratory work, passing the exam for another student; copying materials protected by the copyright system without the permission of the author of the work.

The policy and principles of academic integrity are defined in Section 3 of the Code of Honor of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute". Read more: <https://kpi.ua/code>

Academic Conduct and Ethics Policy

Students should be tolerant, respect the opinions of others, formulate objections in the correct form, and constructively maintain feedback in the classroom.

The norms of ethical behavior of students and employees are defined in Section 2 of the Code of Honor of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute". Read more: <https://kpi.ua/code>

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

Distribution of study time by types of classes and tasks:

Semester	Study time		Distribution of training hours				Control measures		
	Loans	Acad. H.	Lecture	Computer workshop	Lab. Rob.	SRS	FDM	RR	Semester control
2	7	210	36	54	-	120	1	–	Exam

The student's rating in the discipline consists of the points that he receives for:

- performing tasks of a computer workshop and ICR (ICR can be carried out in the form of open tasks, tests or tests on the Sikorsky platform);
- exam (the exam can be conducted in the form of open tasks, or in the form of tests on the Sikorsky platform);
- as well as (at the request of the student) for independent completion of distance courses.

Semester control is an exam.

System of rating (weight) points and evaluation criteria

Completing tasks in laboratory classes and MCR (50 points).

A weight score for each task is given in the Computer Workshop section.

Criteria for evaluating the performance of the task

Completeness and signs of task completion	%
The task is completed in full, all the main and additional data are given	95-100
Minor deficiencies that do not significantly affect the results	80-94
Untimely completion of the task	50-79
Untimely completion of the task, shortcomings	20-49
Poor-quality, untimely completion of the task	Up to 20
Failure to complete the task	0

Answer the exam (50 points).

The ticket contains 2 questions and a problem. The weight score for each question is 15 points. The weight score for the problem is 20 points.

Under the condition of testing, all test tasks have the same weight, the total maximum score is 50.

Criteria for evaluating the answer

Completeness and signs of task completion	%
The task is completed in full, all the main and additional data are given	95-100
Minor deficiencies that do not significantly affect the results	75-94
Incomplete answer, some secondary data is not given. Shortcomings that affect the result, but the student's actions are in the right direction. The student can correct the answer on his own for the sake of correct results.	50-74

<i>Incomplete answer, some basic data is not given. Shortcomings that affect the result, but the student's actions are in the right direction. The student can correct the answer if he uses his own synopsis.</i>	<i>20-50</i>
<i>Poor quality, task performance. The answer contains significant errors. The result is wrong, but the student's actions are in the right direction. The student cannot correct the results without additional help from the teacher and the use of the notes.</i>	<i>Up to 20</i>
<i>Failure to complete the task, lack of response</i>	<i>0</i>

The total number of points in the discipline cannot exceed 100 points.

At the first attestation, the student receives "enrolled" if his current rating is at least half of the maximum possible at the time of the first attestation.

At the second attestation, a student receives "enrolled" if his current rating is at least half of the maximum possible at the time of the second attestation.

The maximum amount of points per semester is 50. A prerequisite for admission to the exam is a rating of at least 10 points.

*To obtain a score, the sum of all received rating points **R** is converted according to the table:*

Score	Score
<i>95...100</i>	<i>Perfectly</i>
<i>85...94</i>	<i>Very good</i>
<i>75...84</i>	<i>well</i>
<i>65...74</i>	<i>Satisfactory</i>
<i>60...64</i>	<i>enough</i>
<i>RD < 60</i>	<i>Disappointing</i>
<i>Not met the conditions of admission</i>	<i>not allowed</i>

9. Additional information on the discipline (educational component)

<https://cpsm.kpi.ua/>

Approximate list of questions that can be submitted to the ICR and to the exam

Computer technical means: indicate the composition of the computer, purpose, external devices.

Compose a batch command file that does the following: makes a copy of prvt.txt in prvt.bak, extracts all text files from the directory.

Give general information about the algorithmic language Fortran and classify it.

What is the purpose of LINK.EXE, LIB.EXE programs?

BIOS: give purpose, progress.

Compose a batch command file that performs the following actions: c1.txt and c2.txt combines the files into a c.txt file and displays its contents on the screen.

Specify the file composition of the Fortran compiler of the specified implementation. Describe the purpose of each file.

Give the definition of the algorithm. What are the known types of algorithms? Give the rules for drawing up flowcharts.

What is a computer self-diagnosis when turning on? Describe the main events and the order of execution.

Compose a batch command file that performs the following actions: compiles the TTT directory and copies all text files from the active directory to it;

Explain what happens when the source code is compiled? Give and describe the stages of compilation.

Explain the purpose and rules of installation: label, comments, hyphenation.

Operating system: give and describe the general scheme of the Win-OS structure.

Compose a batch command file that performs the following actions: Compose a batch command file that performs the following actions: formats the floppy disk and copies Fortran files version 5.0 to it.

Give the composition of the Fortran FTN95 programming system. Describe the purpose of each file.

List the ways to organize cyclical processes. Give examples for each case.

Operating system: give and describe the general scheme of the structure of a single-tasking OS (for example, MS-DOS-OS).

Give a classification of translators. Interpreters and compilers – disadvantages and advantages.

Give ways to organize cycles using conditional control operators. Give examples for each method.

List the functions of the OS and explain them.

Compose a batch command file that does the following: copies c1.txt files and c2.txt to a floppy disk and destroys them on the hard disk.

Give the composition of the Fortran programming system level F77 V3.31. Describe the purpose of each file.

Write according to the rules of Fortran: (task is provided).

Compose a batch command file that performs the following actions: sends the results of the dir program to a text file and displays its contents on the screen.

Give the composition of the Fortran MS F77 V5.0 programming system. Describe the purpose of each file.

Write a program that displays a greeting on the screen: Hello, World!

Give a classification of fixed assets, examples.

Compose a batch command file that performs the following actions: makes copies of all text files, and gives a new extension to the name and destroys the originals.

Describe the general procedure for translating the source code. Give examples.

Explain the purpose, installation and operation of Fortran operators: INTEGER, REAL, COMPLEX, CHARACTER, LOGICAL. Give examples for each of the cases.

Compose a batch command file that does the following: outputs the contents of disk D to a text file and makes a copy of this file on a floppy disk.

What is file typing in OS? What are the types of files and how are they used? Examples.

Compose a batch command file that performs the following actions: makes copies of all text files, and gives a new extension to the name and destroys the originals.

What are compiler passes? What happens in this case? When is the source code syntax check performed? Specify and describe the main console commands of the OS, external and internal commands, how to execute commands, options and parameters of commands. Give examples.

Compose a batch command file that does the following: copies c1.txt and c2.txt files to a floppy disk (drive A) and destroys them on the hard drive.

What is a batch command file and how to automate work with batch command files. Give examples.

What is an ID table, its purpose? Give examples.

What is a console? Describe the features of working in the console mode of the OS.

What types of computer memory are known?

Interpreters and compilers – disadvantages and advantages. Classify Fortran.

What is a console? How to run it? Describe the features of working in console mode.

Give images of blocks of flowcharts, describe the standard capabilities of text editors on the image of blocks of flowcharts.

What software is provided by the manufacturer of the technical means to ensure its full operation?

Compose a batch command file that performs the following actions: formats the floppy disk and copies Fortran files version 3.30 to it.

Describe the procedure for obtaining the *.obj and *.exe module. What are these modules for?

Give examples of user shells and file managers.

Compose a batch command file that performs the following actions: c1.txt and c2.txt combines the files into a c.txt file and displays its contents on the screen.

Give the alphabet and syntax of Fortran. What symbols are used when writing operators and which are used to generate data?

Give a classification of control operators. How is an unconditional transition performed?

What is a computer self-diagnosis when turning on?

Explain the syntax and operation of conditional control operators.

Prioritize arithmetic operations using Fortran.

How are cycles implemented? Give examples of the organization of cyclical processes.

Explain the need for data typing. Give the main types of data, examples of application and explain them.

Specify and explain assignment and default rules for data types. Give examples for each of the cases.

List the functions of the OS and explain them.

Mark up the Fortran screen for a fixed format for placing operators. Explain the location of the operators.

What is the default rule for data types? How is a default rule change done? Give examples.

Explain the purpose and application of the IMPLICIT statement and the PARAMETER statement. Give examples.

What is the dialog mode of the application? Give examples.

Give the definition of the array. Explain the use of static and dynamic data.

Specify the typical structure of the Fortran program.

What is a console? How to get it? Describe the features of working in console mode.

Specify data naming rules: constants and variables, name length, assignment rules.

What are standard Fortran functions? Give examples of functions that are standard and describe the features of the application.

What is an ID table, its purpose? Give examples.

Give and explain the rule for bringing data types. Give examples.

What does a source code developer require to use the F90 extension in the file name of the input code? Explain with an example.

What is the default rule for data types? How is a default rule change done? Give examples.

Compose a batch command file that performs the following actions: sends the results of the dir program to a text file and displays its contents on the screen.

Explain the purpose and application of the DIMENSION statement. Give examples.

Describe the syntax and operation of a conditional arithmetic control operator. Give an example.

Specify data naming rules: constants and variables, name length, assignment rules.

Describe the syntax and operation of the conditional logical control operator. Give an example.

Describe the syntax and operation of the conditional block control operator. Give an example.

Give cycle operators, describe the work. Give examples.

What functions are distinguished in Fortran? Give examples and explain.

What is the area of sight of objects? How do I change the visibility zone of objects?

To give and explain the methods of data transfer between program units.

What software units are used in Fortran? Explain the features.

How do you work with files in the Fortran program?

How is it possible to transfer data between software units?

What are shared memory blocks? Give and explain examples.

How do I transfer data between software units through the parameter list? Explain the rules and features.

What is a file interface? How to implement it?

What are modules used for? Give examples and explain.

In what cases can the list of parameters of a program unit be empty? Explain with examples.

Work program of the educational component (syllabus):

Compiled by Assoc. Prof., Ph.D., Sidorov D.E.

Approved by the Department of KPSM (Minutes No. 17 of 04.06.2025)

Approved by the ICF Methodological Commission (Minutes No. 11 dated 27.06.2025)