Ministry of Education and Science of Ukraine
National Aerospace University
"Kharkiv Aviation Institute"

Department of Aircraft Control Systems (Dep. 301)

APPROVED:
Guarantor of Educational Program
_____ A. S. Kulik
«_27_» _08_____ 2021

WORK PROGRAM OF THE COMPULSORY DISCIPLINE

Fundamentals of Algorithmization and Programming

(name of the discipline)

Field of Study: 17 "Electronics and Telecommunication"

Program Subject Area: 173 "Avionics"

Educational Program: Systems of Autonomous Navigation and Adaptive Control of Aircrafts
(code number and the name of specialization)

Level of Qualification: 1st (bachelor degree)

Kharkiv 2021

Work program of the compulsory discipline "Fundamentals of Algorithmization and Programming" is for English-speaking students of training direction 173 "Avionics".

«27» 08 2021, 10 p.

Developer: O.V. Havrylenko, docent of dep. 301, Candidate of Science (Engineering)

(sign)

The work program has been examined at the meeting of dep. 301 "Aircraft Control Systems".

Record of proceeding: No. 1 from «27» 08 2021

Head of the department
PhD (Engineering), Candidate of Science (sign) K. Yu. Dergachov

1. Course description

<table>
<tr><td colspan="2" rowspan="2">Indices</td><td rowspan="2">Field of study, Program subject area, Educational program</td><td colspan="2">Course specification</td></tr>
<tr><td colspan="2">Full-time study</td></tr>
<tr><td colspan="2">ECTS credits – 13,5</td><td rowspan="3">Branch of Education:
17 – Electronics and Telecommunication</td><td colspan="2">Professional training subject</td></tr>
<tr><td colspan="2">Modules – 2</td><td colspan="2" rowspan="2">Calendar year</td></tr>
<tr><td colspan="2">Semantic modules – 3</td></tr>
<tr><td colspan="2" rowspan="2">Individual Assignment
_____
(topic name)</td><td rowspan="3">Program subject area:
173 – Avionics</td><td colspan="2">2021-2022</td></tr>
<tr><td colspan="2">Semesters</td></tr>
<tr><td>1</td><td>2</td></tr>
<tr><td colspan="2">Total hours – 405</td><td rowspan="4">Educational Program:
Systems of Autonomous Navigation and Adaptive Control of Aircrafts</td><td colspan="2">Lectures</td></tr>
<tr><td colspan="2">Academic hours per day for full-time study</td><td>16</td><td>24</td></tr>
<tr><td colspan="2">Semester 1</td><td colspan="2">Tutorial classes</td></tr>
<tr><td>contact (in class)
2 hrs.</td><td>self-study
12.8 hrs.</td><td>32</td><td>32</td></tr>
<tr><td colspan="2">Semester 2</td><td colspan="2">Lab classes (practical)</td></tr>
<tr><td>3.5 hrs.</td><td>4.9 hrs.</td><td>32</td><td>32</td></tr>
<tr><td colspan="2" rowspan="3"></td><td rowspan="3">Level of qualification:
1st (bachelor degree)</td><td colspan="2">Self-study work</td></tr>
<tr><td>115</td><td>122</td></tr>
<tr><td colspan="2">Assessment form</td></tr>
<tr><td colspan="2"></td><td>Pass</td><td>Exam</td></tr>
</table>

Note: ratio of tutorial and unaided (self-study) work makes: 237/168 (under full-time education).

2. Purpose and objectives of academic discipline

Learning Aims – mastering of methods and means of designing and implementing data processing algorithms, as well as structural approach to software development of computer-aided control systems.

Learning Objectives – studying methods of algorithms designing, learning basic instructions syntax of high-level programming language (C++), as well as getting experience in designing and implementing software for engineering purposes.

Learning Outcomes

According to the requirements of the educational-professional program, students should achieve the following **competencies**:

GC1. Ability to abstract thinking, analysis and synthesis.

GC 2. Ability to apply knowledge in practical situations.

GC 3. Ability to communicate in a foreign language.

GC 5. Ability to learn and master a modern knowledge.

GC 6. Ability to search, process and analyze information from various sources.

PCS2. The ability to use science and technology in the profession, to argue the choice of methods for specialized tasks of analysis and synthesis systems of avionics systems.

PCS 3. The ability to implement and use hardware and algorithmic tools to increase the accuracy and reliability of control systems and other qualities of the aircraft.

PCS 7. The ability to determine the composition of the testing equipment necessary for experiments to determine the characteristics and parameters and control system aircraft.

PCS 9. The ability to introduce achievements of domestic and foreign science and engineering, to use innovative experience in avionics

Program learning outcomes:

PLO1. Use different forms of representation of avionics systems and describe their different methods (verbal, graphic, formally), analyze situations that may occur during their operation

PLO3. Use science and technology in the professional activity, to argue the choice of methods for solving specialized tasks of analysis and synthesis of avionics

PLO4. Apply modern technologies for automation of design and construction of information and control systems in the avionics field, be able to create hardware and software to increase the accuracy, reliability of control systems and other quali-

ties of the aircraft

PLO8. To determine the structure and parameters of the test equipment to conduct experiments to determine the characteristics of the instruments and control systems aircraft, parameters their components and products.

PLO14. Preserve and increase moral, cultural, scientific achievements and values of society by understanding the history and patterns of development of this domain, its place in the overall system knowledge and the development of society, techniques and technologies, use different types and forms of healthy living

Interdisciplinary Relations:

Prerequisites for studying this discipline:

Higher mathematics: calculating the systems of equations, functions research and plotting the graphs construction; vector algebra. Basics of Informatics and using of computers.

The course supports the following courses:

Object-oriented Programming, Fundamentals of Databases, Computational methods and Simulation Techniques.

3. Content of the course

Module 1. Basic algorithms and basic syntactic structures of C ++ language

Semantic modulus 1. Common algorithms for mathematical calculations

THEME 1. Introduction to the discipline

Software development stages. Problem statement. Problem analysis, problem formulation and solution method choice. Software design. Software development. Software modification. Classification of programming languages. Low and high level languages. Compiler. Linker. Basics of projects creating and running in Visual Studio.

THEME 2. Fundamentals of algorithmization and basic concepts of C ++ language

Console Input and Output in C++. Concept of data type. Basic (simple) data types. Integer datatype (int). Symbol datatype (char). Logical datatype (bool). Types with floating point (float, double). Variables. C++ keywords. Constants and literals. Operations: Assignment operations, Arithmetic operations, Comparison operations, Logical operations. Operations priority. Expressions. Data types converting. Preprocessor directives in C++ modules.

Module 2. Advanced algorithms and processing structured data

Semantic modulus 2. Algorithms with branches, repetitions and subroutines

THEME 3. Branching algorithms

Algorithms and forms of their presentation. Algorithms steps. Start and end of the algorithm. Data input and output blocks. Processing block. Decision making structures. Conditional statement (if, if-else). Choice statement (switch). Branching point block.

THEME 4. Loop algorithms

Loop structures. Flowcharts of loop algorithms. Loop with precondition (while). Loop with post-condition (do-while). Loop with parameters (for). Instructions for loops guiding (break, continue).

THEME 5. Arrays processing

One-dimensional arrays. Array elements input and output. Array elements access. Two-dimensional arrays (matrixes). Matrixes input and output. Matrix elements access. Common algorithms with arrays.

THEME 6. Program structuring with subroutines

C ++ pointers and references. Function declaration, definition and calling. The void datatype. Return statement. Function arguments (call by value, call by pointer, call by reference). Local variables and scope. Global variables. Subroutines on flowcharts.

Semantic module 3. Consequent data structures processing

THEME 7. Operating with files and sorting algorithms

File types in terms of programming language. Files processing using C standard library. Files processing using C++ stream library. Sorting algorithms. Selection sort. Insertion sort. Bubble and Shaker sort .

THEME 8. String processing algorithms

Character strings declaration. Using the NULL character. Character string initialization. Strings as function arguments. Standard string datatype. String methods. Search algorithms. Linear search. Binary search. Substring search in a string.

THEME 9. User data types and data structures

Data Structures (records). Defining a Structure. Arrays of structures. Structures as Function Arguments. Returning structure from function. Functions members of structure. Pointers to Structures. The difference between structures and classes. Enumerated Types. Dynamic data structures. Module structure of the program project.

THEME 10. Algorithms analysis and recursive algorithms

The running times of algorithm. Asymptotic notation. Big-θ (Big-Theta) notation. Functions in asymptotic notation. Big-O notation. Recursive algorithms analysis. Simple recursion. Multiple recursion.

4. Course structure

Semantic modules and topics	Hours					
	full-time					
	total	among them				
		lect	pract	lab	ind	self
1	2	3	4	5	6	7
Module 1						
Semantic Modulus 1 – Common algorithms for mathematical calculations						
1. Introduction to the discipline	22	2	0	0	0	20
2. Fundamentals of algorithmization and basic concepts of C ++ language	66	8	0	8	0	50
3. Branching algorithms	45	6	0	8	0	31
Module control	2	0	0	0	0	2
Total for semantic modulus 1	135	16	0	16	0	103

Module 2						
Semantic Modulus 2 – Algorithms with branches, repetitions and subroutines						
4. Loop algorithms	16	2	0	4	0	10
5. Arrays processing	26	4	0	8	0	14
6. Program structuring with subroutines	20	2	0	4	0	14
Module control	2	0	0	0	0	2
Total for semantic modulus 2	64	8	0	16	0	40
Semantic Modulus 3 – Consequent data structures processing						
7. Operating with files and sorting algorithms	18	4	0	4	0	10
8. String processing algorithms	18	4	0	4	0	10
1	2	3	4	5	6	7
9. User data types and data structures	25	6	0	8	0	11
10. Algorithms analysis and recursive algorithms	10	2	0	0	0	6
Module control	2	0	0	0	0	2
Total for semantic modulus 3	71	16	0	16	0	39
Module 2 total	135	24	0	32	0	79
Course total	**270**	**40**	**0**	**48**	**0**	**182**

5. Topics of seminar classes

№ a/o	Topic name	Hours
1	Not appointed	–

6. Topics of practice classes

№ a/o	Topic name	Hours
	Not appointed	

7. Topics of laboratory trainings

№ a/o	Topic name	Hours
1	Basic input-output in C++	4
2	Processing variables of basic data types	5
3	Implementation of algorithms with branches	4
4	Implementation of algorithms with loops	4
5	Arrays processing	4
6	Structuring programs with functions	4
7	Operating with files	4
8	Operating with strings	2
9	Structures processing	2
Module 2 lab classes total		38

8. Self-study (unaided works)

№ a/o	Topic name	Hours
1	*2*	*3*
1	C++ on-line compilers and Visual Studio installing	20
2	Analyzing examples, watching tutorial video about basics of C++ , studying cmath library documentation	50
3	Analyzing examples of branching algorithms, studying flowchart notations	33
4	Studying tutorial books, analyzing examples of iterating algorithms	10
5	Studying tutorial books, analyzing examples and basic array algorithms	14
6	Studying tutorial books, analyzing examples of function definitions	16
7	Studying tutorial books, analyzing different sorting algorithms	10
8	Studying tutorial string library documentation, analyzing examples	10
9	Studying tutorial books, analyzing examples of defining dynamic data structures	11
10	Analyzing examples of recursive algorithms	8
Total hours		118

9. Individual assignments

№ a/o	Topic name	Hours
1	Not appointed	–
Total hours		–

10. Teaching methods

Lectures delivering, laboratory training reports submission, individual consultations (if necessary), independent work of students with the tutorials books, on-line documentation.

11. Forms of control

Current test points that score submitted lab reports and individual assignments, evaluation (grades) of semantic topics, final examination.

12. Appointment of grade points obtaining by a student (credit points)

Semester 1

Components of the study work	Grades for one lesson (task)	Number of lessons (tasks)	Total grades
Semantic modulus 1			
Lecture activity	0…2	8	0…16
Laboratory work implementation and report submission	0...20	3	0...60
Modular and current tests	0…24	1	0…24
Total			**60…100**
Semester 2			
Semantic modulus 2			
Lecture activity	0…1	8	0…8
Laboratory work implementation and report submission	0...10	3	0...30
Modular and current tests	0…12	1	0…12
Semantic modulus 3			
Lecture activity	0…1	4	0…4
Laboratory work implementation and report submission	0...10	3	0...30
Modular and current tests	0…16	1	0…16
Total			**0…100**

Grades scale: Ukrainian and ECTS

Grades	Marks	
	Examination	Pass
90 – 100	excellent	passed
75 – 89	good	
60 -74	satisfied	
0 – 59	unsatisfied	not passed

13. Methodical support

All methodical support is electronically located on a cloud storage and is open to all users. The author of the developments is the Associate professor of the dept. 301 Havrylenko O.V. Link for viewing and downloading:

https://drive.google.com/open?id=0B0v3s_o3YMPmTnlQQ252RTFhb0E

1. Summary of lectures on discipline " Fundamentals of Algorithmization and Programming ". 2021
2. Slides with presentations of lecture materials on discipline " Fundamentals of Algorithmization and Programming ". 2021
3. Methodical instructions and tasks for laboratory work on discipline " Fundamentals of Algorithmization and Programming ". 2021

14. Recommended reading

basic

1. Bjarne Stroustrup. The C++ Programming Language (4th Edition) . – Pearson Education, Inc. – 2013.
2. Stephen Prata, C++ Primer Plus, 6th Edition. – Addison-Wesley Professional. – 2012.
3. C++ : how to program / P.J. Deitel, H.M. Deitel. -- 8th ed. – Pearson Education, Inc. – 2012. – 1303 p.
4. Niklaus Wirth. Algorithms and Data Structures. – Prentice Hall. – 1985. – 288 p.

complementary reading

1. Bjarne Stroustrup. Programming: Principles and Practice Using C++ (2nd Edition). – Pearson Education, Inc. – 2014.
2. Bruce Eckel . Thinking in C++, Volume One: Introduction to Standard C++ (2nd Edition). Prentice Hall – 2000. – 814 p.

15. Information resources

1. https://www.tutorialspoint.com/cplusplus/
2. http://www.cplusplus.com/reference/
3. https://msdn.microsoft.com/en-us/library/cscc687y.aspx
4. https://www.khanacademy.org/computing/computer-science/algorithms