



CS104

PROGRAMMING

Approved: #11 / 11.11.2011.

Updated: #12 / 01.12.2015, #16 / 17.06.2016, #6 / 01.10.2018,
#23 / 28.06.2022

Lecturer: Assoc. Prof. Dr. Dimitar Minchev

ANNOTATION

The discipline provides basic knowledge about computers, programming, programming languages and algorithms. The concepts: source control system, algorithm, integrated development environment, source code, compilation and execution are introduced. The following topics are covered: data types, comments, input and output, variables and constants, operators, checks (conditional *operator*), repetitions (loops), subprograms (procedures *and functions*), arrays, multidimensional arrays and text strings.

MAIN OBJECTIVES AND LEARNING OUTCOMES

The main goal of the course is to master the principles of programming and the basics of the C++ programming language, forming skills for compiling and implementing algorithms. Upon successful completion of the course, students will:

- know the basic data types, variables and constants;
- work with streaming input and output data;
- create branch structures, numeric comparisons, and use Boolean operations;
- implement iterative algorithms using loops;
- create procedures and functions, pass parameters, return value;
- be familiar with the scope of variables;
- work with arrays, multidimensional arrays and text strings.

PREREQUISITES

A necessary prerequisite for the course is preliminary preparation of students in general computer literacy.

STATUS AND STRUCTURE

Specialty	Status	Credits	Full-time study				Part-time study			
			L	S	E	total	L	S	E	total
Informatics and Computer Science	Mandatory	7	30	40		70	20	15	35	
Computer Systems and Technologies	Mandatory	7	30	40		70	20	15	35	
Communications and Electronization for Renewable Energy Sources	Mandatory	7	30	40		70	20	15	35	
Applied Informatics and Multimedia	Mandatory	7	30	40		70	20	15	35	
Software Engineering	Mandatory	7	30	40		70	20	15	35	
Electric Energy Distribution and Electrical Equipment	Mandatory	7	30	40		70	20	15	35	
System Engineering in Industry and Tourism	Mandatory	7	30	40		70	20	15	35	

COURSE CONTENT

1. **Source control.** Version control systems and team interaction.
2. **First steps.** Algorithm. Block diagram. Basic concepts. History of language. Alphabet of the language. Development environment. Compile and run. Comments. Declaration and initialization. Constants and variables.
3. **Calculations.** Integers and floating point numbers. Input and output from the console. Mathematical operations. Remainder of integer division. Separating the digits of a number. Increment and decrement. Mathematical functions. Linear algorithms.
4. **Checks.** Branched algorithms. Conditional operator. Extended syntax. Conditional operation. Comparison operators. Logical expressions. Nested checks.
5. **Repetitions.** Cyclic algorithms. Iterations. Repetition operators with parameter, precondition and postcondition. Nested loops.
6. **Subprograms.** Procedures and functions. Parameters and return value. Scope of variables. Local and global variables. Recursion.
7. **Arrays.** Indexing, traversing, inputting, outputting, and sorting one-dimensional arrays.
8. **Multidimensional arrays.** Matrices. A character string. String functions.

SEMINAR EXERCISES

1. **Source control.** Console work with a distributed source code control system.
2. **First steps.** Introduction to the integrated development environment. Entering the first program. Compiling and running the first program. Writing comments. Concept of input and output. Simple output formatting. Familiarity with constants and variables.
3. **Calculations.** Inputting and outputting integers and floating-point numbers from the console. Calculations of mathematical expressions and use of mathematical functions.
4. **Checks.** Conditional operator usage and its extended syntax. Implementation of nested conditional operators.
5. **Complex checks.** Formation of complex conditions. Logical operations. Checking of input correctness. Checking of alternatives.
6. **Repetitions.** Writing iterative Algorithms using loops. Algorithms with a predetermined number of executions or condition-dependent algorithms.
7. **Repetitions of higher complexity.** More complex and nested loops.
8. **Subprograms.** Functions and Procedures. Declaring and implementing functions. Specifying input and output data. Result return. Parameters by value and address.
9. **Arrays.** One-dimensional arrays: indexing, traversal, input, output and sorting.
10. **Multidimensional arrays.** Problems on matrices, multidimensional arrays and strings. Introduction to string functions.

PLANNED LEARNING ACTIVITIES AND METHODS OF INSTRUCTION

Teaching methods

- Attended lectures and seminar exercises
- Practical education
- Interactive learning
- Visual learning
- E-learning through the Moodle platform

Teaching aids

- Independent work
- Practical tasks
- Solving tasks using application software
- Educational video materials including video presentations
- Use of electronic resources on the Moodle platform in the form of theoretical materials, tests and assignments for independent work on each topic

ASSESSMENT METHODS

Each student's work during the semester is evaluated with a running grade. It is obtained as a result of: coursework, homework, exams, classroom and out-of-classroom attendance of the student.

The final exam is a test with open and closed type questions.

The final grade is formed as a sum of the semester exam and the current grade.

To form the final grade, the student collects points, the maximum value of which is 100, and the distribution of points by evaluated activities is as follows:

1. Auditorial attendance	20 points
1.1. Exams	10 points
1.2. Current control of seminar exercises (<i>visits and participations</i>)	10 points
2. Outside-classroom activities	30 points
2.1. Coursework	10 points
2.2. Homework	20 points
3. Examination procedure	50 points

The student must have collected a minimum of **16** points during the semester and a minimum of **20 points** from the examination procedure. The final grade is formed by distributing the points on the scale:

- from 36 to 50 points = Medium (3);
- from 51 to 65 points = Good (4);
- from 66 to 80 points = Very good (5);
- from 81 to 100 points = Excellent (6).

RECOMMENDED LITERATURE

1. P.Deitel, H.Deitel. C++ How to Program, 10th Edition. Pearson. 2017.
2. Б. Овърленд. C++ на разбираем език. АлексСофт. 2017.
3. Д. Колисниченко. C / C++ - практическо програмиране в примери. Асеновци. 2017.
4. L. Laskov. Programming in C++: Examples and solutions - Part One. NBU. 2016.
5. S. Lippman, J. Lajoie, B. Moo. C++ Primer (5th Edition). Addison-Wesley. 2015.
6. А. Василев. C ++ в примери и задачи. Асеновци. 2015.
7. Б.Йовчева, И.Иванова, П. Петров. Втори стъпки в програмирането на C / C++. КЛМН. 2014.
8. Б.Йовчева, И.Иванова. Първи стъпки в програмирането на C / C++. КЛМН. 2013.
9. B. Stroustrup. The C++ Programming Language, 4th Edition. Pearson. 2013.
10. П. Азълв, Ф. Златарова. C ++ в примери, задачи и приложения. Просвета. 2011.
11. М. Тодорова. Програмиране на C++; ч.1, Сиела, 2010.
12. С. Майерс. По-ефективен C++, 35 нови начина да подобрите своите програми и проекти. ЗеСТ Пресс. 2004.
13. С. Майерс. Ефективен C++ - второ издание: 50 конкретни начина да подобрите своите програми и проекти. ЗеСТ Пресс. 2003.
14. К. Хорстман. Принципи на програмирането със C++, ИК Софтех, София, 2000.

INTERNET RESOURCES

1. М.Тодорова. Увод в програмирането, Структури от данни в програмирането, Обектно ориентирано програмиране. ФМИ, СУ. Източник: <http://informatics.smg.bg/uploads/2015-10-09/magdalena.todorova-cplusplus.programirane.pdf>
2. Е. Петков. Основи на програмирането със C++. В. Търново. 2014. Източник: https://www.researchgate.net/profile/Emiliyan_Petkov/publication/278033468_Osnovi_na_programirane_na_ss_C/links/557adad408aee5c4604491c1/Osnovi-na-programiraneto-ss-C.pdf
3. Интегрирана среда за разработка Code::Blocks. Източник: <http://www.codeblocks.org/downloads/binaries>
4. Онлайн компилатор за програмния език C++. Източник: <https://repl.it/languages/cpp>
5. Е-платформа за онлайн споделяне на програмен код. Източник: <https://codeshare.io/new>
6. Е-справочник на стандартите в програмните езици C и C++. Източник: <http://en.cppreference.com/w/>
7. Е-справочник на библиотечните класове, функции и обекти в програмния език C++. Източник: <http://www.cplusplus.com/>