

**Syllabus**  
**Performance, Data Structures & Algorithms**

<b>1. GENERAL INFORMATION</b>	
Faculty	Information Technology
Major code and title	5B060200 Computer Science
Year, semester	2 <sup>nd</sup> year, 3 <sup>rd</sup> semester
Subject category	Elective
Number of Credits	3
Language of Delivery	English
Prerequisites	Mathematics
Postrequisites	Object-Oriented Programming
Lecturer	Sultan D. Alpar, lecturer, Master of Engineering Sciences Room 808, e-mail: <a href="mailto:rapla.natlus@gmail.com">rapla.natlus@gmail.com</a> Office hours: Tue, Wed, Fri 11.00-13.00
Instructors	Sultan D. Alpar
<b>2. GOAL AND OBJECTIVES OF THE COURSE</b>	
<p>The goal of the course is learning the basics of algorithms, data structures and programming to solve practical problems using the language C++.</p> <p>The objectives of the course are:</p> <ul style="list-style-type: none"> <li>• studying of the rules of formulating and solving practical problems on a computer</li> <li>• introduction to the basics of the theory of algorithms</li> <li>• to study how to present the program and methods of data processing in the computer</li> <li>• to study the structure of programming systems and rules for using in applications development</li> <li>• mastering skills of programming using high-level language C++ to solve practical problems on a computer</li> <li>• development of programs with use of standard modules, style of programming, methods of testing</li> <li>• to use the algorithms for solving typical practical tasks</li> </ul> <p style="text-align: center;"><b>Learning outcomes of the course</b></p> <p>Students successfully completing the course will be able to:</p> <ul style="list-style-type: none"> <li>• To develop block diagrams of various algorithms</li> <li>• To organize depending on task requirements necessary structures of the data</li> <li>• To develop programs on C++ with use of means of language</li> <li>• To write programs in good style</li> <li>• To debug and test programs</li> <li>• To constitute the qualitative program documentation</li> </ul>	
<b>3. COURSE DESCRIPTION</b>	
<p>The course "Performance, Data Structures &amp; Algorithms" is designed for studying of development of algorithms and programs for solutions of different problems. For this purpose, subjects of this course are considered, such as program structure, principles of construction of algorithms and program, methods of solving, algorithmization, programming, debugging and implementation of programs, using the language C++.</p>	
<b>4. COURSE POLICY</b>	
<p><b>Students are forbidden to:</b></p> <ul style="list-style-type: none"> <li>- submit any tasks after the deadline. Late submissions are graded down.</li> <li>- cheat. Plagiarized papers shall not be graded;</li> </ul>	

- be late for classes;
- retake any tests, unless there is a valid reason for missing them;
- use mobile phones in class.

**Students should always**

- be appropriately dressed (formal/semi- formal styles are acceptable);
- show consideration for and mutual support of teachers and other students;
- let the teacher know of any problems arising in connection with their studies.

**5. Literature**

Basic literature:

1. H. M. Deitel, Prentice Hall, C++, How to Program, 5th.Edition.
2. Bruce Eckel, Thinking in C++, Volume 1, 2nd Edition
3. N. Kultzin, C/C++ in exercises and examples.

Supplementary literature:

1. <http://www.doc.ic.ac.uk>
2. <http://www.cplusplus.com>
3. <http://cppstudio.com>

**6. Course Content**

**6.1 Lecture, practical/seminar/laboratory session plans**

Abbreviation	Meaning
TSIS	Teacher supervised independent work (CPCII)
SIS	Students' independent work (CPC)
IHW	Individual homework
PA	Practical assignment

Week No	Course Topic	Reference Materials	Lectures (1 h/w)	Practical classe (1 h/w)	TSIS (1 h/w)	SIS (3 h/w)
1	<b>History of C++. Variables and Types. Flowcharts. Building blocks.</b> Supported paradigms. The Features of C++ as a Language. Visual Studio. Support for C++. Console Appication. Structure of a program. Comments. Using namespace std. Variables and Types. Identifiers. Fundamental data types. Character types. Numerical integer types. Floating-point types. Boolean type. Basic mathematical functions in the C++ mathematics library	Basic [1]-[3]	L 1	PA 1	TSIS 1	SIS 1

2	<b>Declaration of variables. Operators.</b> Initialization of variables. Type deduction: auto and decltype. Introduction to strings. Literals. Integer Numerals. Floating Point Numerals. Character and string literals. Other literals. Typed constant expressions. Preprocessor definitions. Operators. Assignment operator. Arithmetic operators. Compound assignment. Increment and decrement. Relational and comparison operators. Logical operators. Conditional ternary operator. Comma operator. Bitwise operators. Explicit type casting operator. Other operators. Basic Input/Output. Standard output. Standard input.	Basic [1]-[3]	L 2	PA 2	TSIS 2	SIS 2
3	<b>Iteration statements (loops).</b> The standard header <code>&lt;sstream&gt;</code> . Statements and flow control. Selection statements: if and else. Iteration statements (loops). Iteration statements (loops). The do-while loop. The for loop. Range-based for loop. Jump statements. The break statement. The continue statement. The goto statement. The switch statement.	Basic [1]-[3]	L 3	PA 3	TSIS 3	SIS 3
4	<b>Arrays.</b> Initializing arrays. Accessing the values of an array.	Basic [1]-[3]	L 4	PA 4	TSIS 4	SIS 4
5	<b>Multidimensional arrays.</b> Arrays as parameters. Library arrays.	Basic [1]-[3]	L 5	PA 5	TSIS 5	SIS 5
6	<b>Multidimensional arrays.</b> Sum of the elements of all matrix. Product of the elements. Number of the elements. Sum of every row and every column.	Basic [1]-[3]	L 6	PA 6	TSIS 6	SIS 6
7	<b>Character sequences.</b> Initialization of null-terminated character sequences. Strings and null-terminated character sequences.	Basic [1]-[3]	L 7	PA 7	TSIS 7	SIS 7
8	<b>Functions.</b> Functions with no type. The use of void. The return value of main. Arguments passed by value and by reference..	Basic [1]-[3]	L 8	PA 8	Mid-term	
9	<b>Functions.</b> Efficiency considerations and const references. Inline functions. Default values in parameters. Declaring functions	Basic [1]-[3]	L 9	PA 9	TSIS 8	SIS 8
10	<b>Recursion. Recursive function.</b>	Basic [1]-[3]	L 10	PA 10	TSIS 9	SIS 9
11	<b>Data structures.</b> Nesting structures.	Basic [1]-[3]	L 11	PA 11	TSIS 10	SIS 10

12	<b>Pointers.</b> Address-of operator. Dereference operator. Declaring pointers. Pointers and arrays. Pointer initialization. Pointer arithmetics. Pointers and const. Pointers and string literals. Pointers to pointers. void pointers. Invalid pointers and null pointers. Pointers to functions.	Basic [1]-[3]	L 12	PA 12	TSIS 11	SIS 11
13	<b>Files.</b> Input/output with files. Open a file Closing a file.	Basic [1]-[3]	L 13	PA 13	TSIS 12	SIS 12
94	<b>Pointers and arrays. Sorting.</b>	Basic [1]-[3]	L 14	PA 14	TSIS 13	SIS 13
15	<b>Sorting.</b>	Basic [1]-[3]	L 15	PA 15	End of term	
<b>Total hours</b>		<b>90</b>	<b>15</b>	<b>15</b>	<b>15</b>	<b>45</b>

## 6.2 List of assignments for Student Independent Study

№	Assignments (topics) for Independent study	Hours	Recommended literature and other sources (links)	Form of submission
1	Fundamental data types. Boolean type. Basic mathematical functions in the C++ mathematics library.	1	Basic [1]-[3] Supplementary [1][2]	Discussion
2	Operators. Assignment operator. Arithmetic operators. Basic Input/Output.	1	Basic [1]-[3] Supplementary [1][2]	Discussion
3	Iteration statements (loops). The do-while loop. The for loop. Range-based for loop. Jump statements. The break statement. The continue statement. The goto statement. The switch statement.	1	Basic [1]-[3] Supplementary [1][2]	Discussion
4	One-dimensional arrays.	1	Basic [1]-[3] Supplementary [1][2]	Discussion
5	Multidimensional arrays.	1	Basic [1]-[3] Supplementary [1][2]	Discussion
6	Tasks with Multidimensional arrays. Sum of the elements of all matrix. Product of the elements. Number of the elements. Sum of every row and every column.	1	Basic [1]-[3] Supplementary [1][2]	Discussion
7	Character sequences. String.	1	Basic [1]-[3] Supplementary [1][2]	Discussion
8	Function.	1	Basic [1]-[3] Supplementary [1][2]	Discussion
9	Recursivity.	1	Basic [1]-[3] Supplementary [1][2]	Discussion
10	Function. Arguments passed by	1	Basic [1]-[3]	Discussion

	value and by reference		Supplementary [1][2]	
11	Data structures	1	Basic [1]-[3] Supplementary [1][2]	Discussion
12	Pointers	1	Basic [1]-[3] Supplementary [1][2]	Discussion
13	Files	1	Basic [1]-[3] Supplementary [1][2]	Discussion

### 7. Student performance evaluation system for the course

Period	Assignments	Number of points	Total
1 <sup>st</sup> attestation	<b>Class work:</b> Regular attendance\Active participation Quiz # 1 Quiz # 2 <b>Student Independent Study:</b> Home Work Assignments <b>Mid term</b>	<b>20</b> 10 5 5 <b>30</b> 15 15 <b>50</b>	100
2 <sup>nd</sup> attestation	<b>Class work:</b> Regular attendance\Active participation Quiz # 1 Quiz # 2 <b>Student Independent Study:</b> Home Work Assignments <b>End of term</b>	<b>20</b> 10 5 5 <b>30</b> 15 15 <b>50</b>	100
Final exam	<b>Exam</b>	<b>100</b>	100
<b>Total</b>	<b>0,3*1stAtt+0,3*2ndAtt+0,4*Final</b>		<b>100</b>

\*If the number of absences exceeds 20%, student will be automatically scheduled for a Retake (summer semester)

Achievement level as per course curriculum shall be assessed according to the evaluation chart adopted by the academic credit system:

Letter Grade	Numerical equivalent	Percentage	Grade according to the traditional system
A	4,0	95-100	Excellent
A-	3,67	90-94	
B+	3,33	85-89	Good
B	3,0	80-84	
B-	2,67	75-79	
C+	2,33	70-74	Satisfactory
C	2,0	65-69	
C-	1,67	60-64	
D+	1,33	55-59	
D	1,0	50-54	Fail
F	0	0-49	

### 8. METHODOLOGICAL GUIDELINES

Assessment is administered continuously throughout the course. The students are rated against their performance in **continuous rating** administered throughout the semester (credited 60%) and **summative rating** done during the examination session (credited 40%), total **100%**. **Continuous rating** is students' on-going performance in class and independent work. Class work is assessed for attendance and active participation (problem solving).

### **Teaching methodology**

Theory classes:

- lectures developing the theoretical aspects of the subject
- practical classes aimed at applying theory to problems.

Workshop classes:

- practical classes in which students solve problems in groups or individually.

**SIS (Student Independent Study)** comprises topics related problems to be done by students independently and checked in class.

**TSIS (Teacher Supervised Student Independent Study)** comprises individual homework assignments to be done by students independently and checked by teacher.

**Mid-term** examination is held in the 8<sup>th</sup> week of the semester and includes topics 1-7 of the course.

**End-of-term** examination is held in the last week of the semester and includes topics 8-15 of the course.

**Final examination** is a computer-based test that consists of multiple choice questions covering all topics of the course.