



SCHOOL OF BUSINESS
AND MANAGEMENT OF
TECHNOLOGY OF BSU



Innovative ICT Education for Social-Economic Development (IESED)
574283-EPP-1-2016-1-LT-EPPKA2-CBHE-JP

PRINCIPLES OF ALGORITHMIZATION AND PROGRAMMING

Minsk 2017

1. COURSE PLAN

Year of study	Semester	Academic hours					Hours of course work	ECTS	Number of hours
		Total	Lectures	Lab	Practice/ seminar	Independent work			
1	1,2	190	52	68	-	70	-	7	Full-time
1	1,2	190	16	20	-	154	-	7	Part-time

2. COMPETENCIES

1. Develop data structures for use in information systems, operational analysis systems and intellectual systems.
2. Perform modeling, design of software tools and documentation to support activities in various subject areas.
3. Perform comprehensive testing of the developed software products and applied software.

3. COURSE GOAL

Training of a specialist with fundamental knowledge and practical skills in the field of algorithms and programming.

4. COURSE OUTCOMES

After completing this course student will be able to:

- develop data structures for solving applied problems.
- develop algorithms to solve real world problems.
- make use of software development applications for designing systems.
- apply appropriate testing and debugging techniques.

5. COURSE CONTENT (for full-time education)

Section number	Topics, classes; list of issues to be studied	Number of academic hours				Form of knowledge control
		Lecture	Practical classes (seminar)	laboratory classes	Independent work	
1.	Fundamentals of Algorithmization	3	3	0	0	Final examination
1.1.	The concept and properties of the algorithm.	1	1			
1.2.	Methods of describing algorithms	1	1			
1.3.	Stages of program development	1	1			
2.	C++ language basics	18,5	6,5	12	0	Final examination
2.1.	C++.Basic syntax	0,5	0,5			
2.2.	Data types in C++	1	1			
2.3.	Variables, constants in C++	0,5	0,5			

2.4.	Structure of the source code in C++.	0,5	0,5			
2.5.	Operations in C++	1	1			
2.6.	Standard I / O functions in C++.	3	1	2		
2.7.	Operators in C++	12	2	10		Laboratory work
	C# language basics	18,5	6,5	12	0	Final examination
3.1.	C#. Basic syntax	0,5	0,5			
3.2.	Data types in C#	1	1			
3.3.	Structure of the source code in C#.	0,5	0,5			
3.4.	Variables, constants in C#	0,5	0,5			
3.5.	Operations	1	1			
3.6.	Using FCL I / O classes.	3	1	2		
3.7.	Operators in C#	12	2	10		Laboratory work
	Structured Data Types	28	10	18	0	Final examination
4.1.	Arrays in C++	3	1	2		
4.2.	Structures in C++	3	1	2		
4.3.	Array Class C#	3	1	2		
4.4.	Structures in C #	3	1	2		
4.5.	Algorithms for processing arrays.	16	6	10		Laboratory work
	Pointers and Links in C++	7	3	4	0	Final examination
5.1.	Pointers	1	1			
5.2.	The allocation of dynamic memory.	6	2	4		
	Functions	14	6	8	0	Final examination
6.1.	Functions in C++.	1	1			
6.2.	Passing parameters to a function in C++.	3	1	2		
6.3.	Functions and procedures in C#	1	1			
6.4.	Passing parameters to a function in C#.	3	1	2		
6.5.	Recursive calls and their applications.	6	2	4		Laboratory work
	Files and streams	18	4	14	0	Final examination
7.1.	The concept of a file and stream in C++.	1	1			
7.2.	Standard functions for working with files in C++.	3	1	2		
7.3.	The concept of stream in C#.	3	1	2		
7.4.	Using stream classes.	3	1	4		Laboratory work
7.5.	Algorithms files processing.	6		6		Laboratory work
	String type	15	5	10	0	Final examination
8.1.	Arrays of symbols in C++	1	1			
8.2.	C++ functions of processing of character arrays	3	1	2		
8.3.	Using class String in C#.	3	1	2		
8.4.	Algorithms for processing strings	8	2	6		Laboratory work
	Classes and objects in C++	23	9	14	0	Final examination
9.1.	Objective paradigm	1	1			
9.2.	Classes	6	2	4		Laboratory work
9.3.	Inheritance	6	2	4		Laboratory work
9.4.	Dynamic objects.	4	2	2		Laboratory work

9.5.	Virtual methods.	6	2	4		Laboratory work
	Dynamic data structures	37	11	26	0	
10.1.	General principles of the organization of dynamic structures in C++.	1	1			
10.2.	Algorithms for Bi-directional lists processing.	10	2	8		Laboratory work
10.3.	Algorithms for Binary trees processing.	14	4	10		Laboratory work
10.4.	Algorithms for Graphs processing.	12	4	8		Laboratory work
	Class and function templates in C++	8	2	6	0	Final examination
11.1.	Function templates.	1	1			
11.2.	Class templates.	7	1	6		Laboratory work
	TOTAL	190	66	124	0	

6. COURSE CONTENT (for part-time education)

Section number	Topics, classes; list of issues to be studied	Number of academic hours				Form of knowledge control
		Lecture	Practical classes (seminar)	laboratory classes	Independent work	
1.	Fundamentals of Algorithmization	3	0,3	0	2,7	Final examination
1.1.	The concept and properties of the algorithm.	1	0,1		0,9	
1.2.	Methods of describing algorithms	1	0,1		0,9	
1.3.	Stages of program development	1	0,1		0,9	
2.	C++ language basics	18,5	1,4	2	15,1	Final examination
2.1.	C++.Basic syntax	0,5	0,1		0,4	
2.2.	Data types in C++	1	0,2		0,8	
2.3.	Variables, constants in C++	0,5	0,1		0,4	
2.4.	Structure of the source code in C++.	0,5	0,1		0,4	
2.5.	Operations in C++	1	0,2		0,8	
2.6.	Standard I / O functions in C++.	3	0,2		2,8	
2.7.	Operators in C++	12	0,5	2	9,5	Laboratory work
3.	C# language basics	18,5	1,4	2	17,1	Final examination
3.1.	C#. Basic syntax	0,5	0,1		0,4	
3.2.	Data types in C#	1	0,2		0,8	
3.3.	Structure of the source code in C#.	0,5	0,1		0,4	
3.4.	Variables, constants in C#	0,5	0,1		0,4	
3.5.	Operations	1	0,2		0,8	
3.6.	Using FCL I / O classes.	3	0,2		2,8	
3.7.	Operators in C#	12	0,5	2	11,5	Laboratory work
4.	Structured Data Types	28	3,2	4	20,8	Final examination
4.1.	Arrays in C++	3	0,5		2,5	

4.2.	Structures in C++	3	0,1		2,9	
4.3.	Array Class C#	3	0,5		2,5	
4.4.	Structures in C #	3	0,1		2,9	
4.5.	Algorithms for processing arrays.	16	2	2	10	Laboratory work
5.	Pointers and Links in C++	7	0,3	0	6,7	Final examination
5.1.	Pointers	1	0,1		0,9	
5.2.	The allocation of dynamic memory.	6	0,2		5,8	
6.	Functions	14	1,6	2	10,4	Final examination
6.1.	Functions in C++.	1	0,1		0,9	
6.2.	Passing parameters to a function in C++.	3	0,2		2,8	
6.3.	Functions and procedures in C#	1	0,1		0,9	
6.4.	Passing parameters to a function in C#.	3	0,2		2,8	
6.5.	Recursive calls and their applications.	6	1	2	3	Laboratory work
7.	Files and streams	18	1,7	2	14,3	Final examination
7.1.	The concept of a file and stream in C++.	1	0,2		0,8	
7.2.	Standard functions for working with files in C++.	3	0,2		2,8	
7.3.	The concept of stream in C#.	3	0,1		2,9	
7.4.	Using stream classes.	3	0,2		4,8	Laboratory work
7.5.	Algorithms files processing.	6	1	2	3	Laboratory work
8.	String type	15	1,5	2	11,5	Final examination
8.1.	Arrays of symbols in C++	1	0,1		0,9	
8.2.	C++ functions of processing of character arrays	3	0,2		2,8	
8.3.	Using class String in C#.	3	0,2		2,8	
8.4.	Algorithms for processing strings	8	1	2	5	Laboratory work
9.	Classes and objects in C++	23	1,2	2	19,8	Final examination
9.1.	Objective paradigm	1	0,2		0,8	
9.2.	Classes	6	0,1	1	4,9	Laboratory work
9.3.	Inheritance	6	0,2	1	4,8	Laboratory work
9.4.	Dynamic objects.	4	0,2		3,8	Independent work
9.5.	Virtual methods.	6	0,5		5,5	Independent work
10.	Dynamic data structures	37	3,1	6	27,9	
10.1	General principles of the organization of dynamic structures in C++.	1	0,1		0,9	
10.2	Algorithms for Bi-directional lists processing.	10	1	2	7	Laboratory work
10.3	Algorithms for Binary trees processing.	14	1	2	11	Laboratory work
10.4	Algorithms for Graphs processing.	12	1	2	9	Laboratory work
11.	Class and function templates in C++	8	0,3	0	7,7	Final examination
11.1	Function templates.	1	0,1		0,9	
11.2	Class templates.	7	0,2		6,8	Independent work
	TOTAL	190	16	20	154	

7. THEORETICAL CONTENT

№	Names of topics	Content
1.	Fundamentals of Algorithmization	
1.1.	The concept and properties of the algorithm.	Algorithm definition. Understanding of algorithm properties.
1.2.	Methods of describing algorithms	Verbal-mathematical description. Flowchart Algorithmic language. Basic principles of drawing up flowcharts.
1.3.	Stages of program development	Compilation. Executable code. Errors classification. Debugging. Programming paradigms.
2.	C++ language basics	
2.1.	C++.Basic syntax	Alphabet, lexemes and separators, Naming.
2.2.	Data types in C++	Concept, classification, description, compatibility of types, conversion.
2.3.	Variables, constants in C++	Variables: description, initialization. Literal constants. Lifetime, scope. Const keyword.
2.4.	Structure of the source code in C++.	Source code structure. Preprocessor understanding. Directives. Statement.
2.5.	Operations in C++	Concept, classification, priority, standard operations. Branching operators.
2.6.	Standard I / O functions in C++.	Formatting input/output functions. String, characters input/output functions.
2.7.	Operators in C++	Branch, cycling operators. Iterative calculations. Iterative calculations with specified accuracy.
3.	C# language basics	
3.1.	C#. Basic syntax	Alphabet, lexemes and separators.
3.2.	Data types in C#	concept, classification, description, compatibility of types
3.3.	Structure of the source code in C#.	Source code structure. Statement. Modules.
3.4.	Variables, constants in C#	Variables: description, initialization. Literal constants. Lifetime, scope.
3.5.	Operations	concept, classification, priority, standard operations. Branching operators. Iterative calculations with specified accuracy
3.6.	Using FCL I / O classes.	Console input/output classes. Methods.
3.7.	Operators in C#	Branch, cycling operators. Iterative calculations. Iterative calculations with specified accuracy.
4.	Structured Data Types	
4.1.	Arrays in C++	Concept, declaration, initialization. Access to the elements of the array. Enumerators.
4.2.	Structures in C++	Architecture of structural type and features of use in the program.
4.3.	Array Class C#	Concept, declaration, initialization. Access to the elements of the array. Enumerators.
4.4.	Structures in C #	Architecture of structural type and features of use in the program.
4.5.	Algorithms for processing arrays.	Input/output of arrayed data. Array search algorithms. Algorithms for sorting arrays. 2D-arrays processing algorithms.
5.	Pointers and Links in C++	
5.1.	Pointers	Concept, declaration, initialization. Operations with pointers.
5.2.	The allocation of dynamic memory.	Create, destroy, reallocation operation and function. References to arrays. Dynamic arrays processing.
6.	Functions	
6.1.	Functions in C++.	Declaration and implementation. Return value from function.
6.2.	Passing parameters to a function in C++.	Declaration and transfer to the function of parameter-values and parameters-references
6.3.	Functions and procedures in C#	Declaration and implementation. Return value from function.
6.4.	Passing parameters to a function in C#.	Declaration and transfer to the function of parameter-values and parameters-references Use of service properties ref, out.

6.5.	Recursive calls and their applications.	Method of recursive programming. Applied use of recursive algorithms.
7.	Files and streams	
7.1.	The concept of a file and stream in C++.	Types of files, features of the structure.
7.2.	Standard functions for working with files in C++.	Organization of input-output with the help of stream-classes cin and cout.
7.3.	The concept of stream in C#.	The namespace System.IO.
7.4.	Using stream classes.	Input / output of character data. I / O of binary data. Random Access Files. MemoryStream class.
8.	String type	
8.1.	Arrays of symbols in C++	Concept, declaration, initialization.
8.2.	C++ functions of processing of character arrays	Standard library function for processing strings.
8.3.	Using class String in C#.	Working with strings in .Net, the basis of regular expressions.
8.4.	Algorithms for processing strings	Searching in text algorithms. Splitting text replacement of text lexemes.
8.5.	Algorithms files processing.	Algorithms for reading, writing and modifying files.
9.	Classes and objects in C++	
9.1.	Objective paradigm	Encapsulation, inheritance, polymorphism.
9.2.	Classes	Defining classes. Fields, methods, access modifiers.
9.3.	Inheritance	Inheritance of classes. Inheritance modifiers. Inheritance overlapping methods and fields
9.4.	Dynamic objects.	Constructors, destructors. this pointer.
9.5.	Virtual methods.	Abstract classes. Virtual methods. Inheriting of virtual methods. Using abstract classes.
10.	Dynamic data structures	
10.1.	General principles of the organization of dynamic structures in C++.	Dynamic structure nodes. Types of dynamic structures.
10.2.	Algorithms for Bi-directional lists processing.	Algorithms for the formation of a list, insertion, deletion, permutation and retrieval of elements.
10.3.	Algorithms for Binary trees processing.	Algorithms for the formation of a search binary tree. Insert, delete, and search for nodes algorithms. Algorithms for traversing a tree. Solution of applied problems using binary trees.
10.4.	Algorithms for Graphs processing.	Algorithms of graph formation. Algorithms for traversing a graph, searching for a path. Solution of applied problems using graphs.
11.	Class and function templates in C++	
11.1.	Function templates.	Template polymorphism understanding. Defining template function. Using template function.
11.2.	Class templates.	Defining template classes. Instanting template classes.

8. LABORATORY PRACTICE

№	Names of topics	Content
2.	C++ language basics	
2.6.	Standard I / O functions in C++.	Creating simple program for linear calculation with formatting output.
2.7.	Operators in C++	Creation of programs for the implementation of branching and cyclic algorithms. Iterative calculations with specified accuracy.
3.	C# language basics	
3.6.	Using FCL I / O classes.	Creating simple program for linear calculation with formatting output.
3.7.	Operators in C#	Creation of programs for the implementation of branching and cyclic algo-

		rithms. Iterative calculations with specified accuracy.
4.	Structured Data Types	
4.1.	Arrays in C++	Implementing Array Elements I / O Algorithms.
4.2.	Structures in C++	Modeling structure data to store table information.
4.3.	Array Class C#	Implementing Array Elements I / O Algorithms.
4.4.	Structures in C #	Modeling structure data to store table information.
4.5.	Algorithms for processing arrays.	Implementing search, sorting algorithms to solve application tasks. Algorithms. Using 2D arrays to model 2D objects.
5.	Pointers and Links in C++	
5.2.	The allocation of dynamic memory.	Working with dynamic arrays.
6.	Functions	
6.2.	Passing parameters to a function in C++.	Create and use of function.
6.4.	Passing parameters to a function in C#.	Create and use of function.
6.5.	Recursive calls and their applications.	Using a recursive algorithm for a two-dimensional maze.
7.	Files and streams	
7.2.	Standard functions for working with files in C++.	Development of a user interface for working with files.
7.3.	The concept of stream in C#.	Development of a user interface for working with files.
7.4.	Using stream classes.	Working with FCL stream classes.
7.5.	Algorithms files processing.	Implementation of various structures data storage in a file.
8.	String type	
8.2.	C++ functions of processing of character arrays	Performing simple character-by-character operations with string data.
8.3.	Using class String in C#.	Performing simple character-by-character operations with string data.
8.4.	Algorithms for processing strings	Converting files with the source code of the program in C ++
9.	Classes and objects in C++	
9.2.	Classes	Developing 2D geometric shapes classes.
9.3.	Inheritance	Developing 3D geometric shapes classes by expanding the classes of 2D figures.
9.4.	Dynamic objects.	Development of dynamic structures for storing instances of 3D shapes classes.
9.5.	Virtual methods.	Implementation of standard operations on various 3D objects.
10.	Dynamic data structures	
10.1.	General principles of the organization of dynamic structures in C++.	Creating bidirectional list class
10.2.	Algorithms for Bi-directional lists processing.	Solution of application tasks using bidirectional class list.
10.3.	Algorithms for Binary trees processing.	Processing binary trees using recursive algorithms
10.4.	Algorithms for Graphs processing.	Solution of application tasks using graphs.
11.	Class and function templates in C++	
11.2.	Class templates.	Create and use template class

9. ASSIGNMENT FOR INDEPENDENT WORK

№	Tasks
1.	Testing in VS environment
2.	Building a multi-module project
3.	Realization of iterative calculations with control of input values and iterative process.
4.	Algorithms for rapid sorting of one-dimensional arrays.
5.	Algorithms for non-standard extraction of elements in two-dimensional arrays.
6.	Organization of storage of the table on the basis of structures.
7.	Recursive processing of a two-dimensional array.
8.	Splitting lines.
9.	Sorting files.
10.	Development of the "unidirectional list" class.
11.	Development of the class "stack"
12.	Development of the "queue" class using templates
13.	Forming a table output using the cout class.
14.	Reading from a file using stream classes

10. SYSTEM OF ASSESSMENT OF KNOWLEDGE AND SKILLS (ACCORDING TO THE NATIONAL REQUIREMENTS)

The evaluation criteria of the results on a ten-point scale.

A ten-point scale, depending on the grade and the mark, includes the following criteria:

10 (ten) points, passed:

- systematized, deep and full knowledge on all sections of the curriculum of the institution of higher education in the academic discipline, as well as on major issues that go beyond its limits;
- accurate use of scientific terminology (including in a foreign language), competent, logically correct statement of the answer to questions;
- perfect mastering of the tools of the academic discipline, the ability to use it effectively in formulation and solution of scientific and professional problems;
- the expressed ability independently and creatively to solve complex problems in non-standard situations;
- complete and profound studying of basic, additional literature on the subject of the discipline;
- the ability to freely navigate in theories, concepts and directions on the discipline and give them an analytical assessment, use the scientific achievements of other disciplines;
- creative independent work on practical, laboratory classes, active creative participation in group discussions, high level of the culture of performance of tasks.

9 (nine) points, passed:

- systematized, deep and full knowledge on all sections of the curriculum of the institution of higher education on the academic discipline;
- accurate use of scientific terminology (including in a foreign language), competent, logically correct statement of the answer to questions;
- mastering of the tools of the academic discipline, the ability to use it effectively in formulation and solution of scientific and professional problems;
- ability independently and creatively to solve complex problems in non-standard situations within the curriculum of the institution of higher education on the academic discipline;
- complete studying of basic, additional literature on the subject of the discipline, recommended by the curriculum of the institution of higher education on the discipline;
- the ability to navigate in theories, concepts and directions on the discipline and give them an analytical assessment;

- Systematic, active independent work on practical, laboratory classes, active creative participation in group discussions, high level of the culture of performance of tasks.

8 (eight) points, passed:

- systematized, deep and full knowledge on all sections of the curriculum of the institution of higher education in the academic discipline in the volume of the curriculum of the institution of higher education on the discipline;
- use of scientific terminology (including in a foreign language), competent, logically correct statement of the answer to questions, the ability to make sound conclusions and generalizations;
- mastering of the tools of the academic discipline (methods of complex analysis, information technology), the ability to use it effectively in formulation and solution of scientific and professional problems;
- ability independently to solve complex problems within the curriculum of the institution of higher education on the academic discipline;
- studying of basic, additional literature, recommended by the curriculum of the institution of higher education on the discipline;
- the ability to navigate in theories, concepts and directions on the discipline and give them an analytical assessment;
- active independent work on practical, laboratory classes, systematic participation in group discussions, high level of the culture of performance of tasks.

7 (seven) points, passed:

- systematized, deep and full knowledge on all sections of the curriculum of the institution of higher education on the academic discipline;
- use of scientific terminology (including in a foreign language), competent, logically correct statement of the answer to questions, the ability to make sound conclusions and generalizations;
- mastering of the tools of the academic discipline, the ability to use it effectively in formulation and solution of scientific and professional problems;
- free possession of generic solutions within the curriculum of the institution of higher education on the academic discipline;
- studying of basic, additional literature, recommended by the curriculum of the institution of higher education on the discipline;
- the ability to navigate in basic theories, concepts and directions on the discipline and give them an analytical assessment;
- independent work on practical, laboratory classes, participation in group discussions, high level of the culture of performance of tasks.

6 (six) points, passed:

- sufficiently full and systematized knowledge in the volume of the curriculum of the institution of higher education on the discipline;
- use of the necessary scientific terminology, competent, logically correct statement of the answer to questions, the ability to make sound conclusions and generalizations;
- mastering of the tools of the academic discipline, the ability to use it effectively in solution of scientific and professional problems;
- ability independently to apply generic solutions within the curriculum of the institution of higher education on the academic discipline;
- studying of basic literature, recommended by the curriculum of the institution of higher education on the discipline;

- the ability to navigate in basic theories, concepts and directions on the discipline and give them a comparative assessment;
- active independent work on practical, laboratory classes, periodic participation in group discussions, high level of the culture of performance of tasks.

5 (five) points, passed:

- sufficient knowledge in the volume of the curriculum of the institution of higher education on the discipline;
- use of scientific terminology, competent, logically correct statement of the answer to questions, the ability to make sound conclusions;
- mastering of the tools of the academic discipline, the ability to use it in solution of scientific and professional problems;
- ability independently to apply generic solutions within the curriculum of the institution of higher education on the academic discipline;
- studying of basic literature, recommended by the curriculum of the institution of higher education on the discipline;
- the ability to navigate in basic theories, concepts and directions on the discipline and give them a comparative assessment;
- active independent work on practical, laboratory classes, periodic participation in group discussions, high level of the culture of performance of tasks;
- independent work on practical, laboratory classes, periodic participation in group discussions, sufficient level of the culture of performance of tasks.

4 (four) points, passed:

- sufficient knowledge within the educational standard of higher education;
- studying of basic literature, recommended by the curriculum of the institution of higher education on the discipline;
- use of scientific terminology, logical statement of the answer to questions, the ability to make sound conclusions;
- ability to draw conclusions without essential errors;
- mastering of the tools of the academic discipline, the ability to use it in solution of standard (typical) tasks;
- ability to solve standard (typical) tasks under the guidance of a teacher;
- ability to navigate in basic theories, concepts and directions on the discipline and give them an assessment;
- work under the guidance of a teacher on practical, laboratory classes, the permissible level of the culture of performance of tasks.

3 (three) points, failed:

- insufficient knowledge within the educational standard of higher education;
- studying of basic literature, recommended by the curriculum of the institution of higher education on the discipline;
- knowledge of a part of the basic literature, recommended by the curriculum of the institution of higher education on the discipline;
- use of scientific terminology, presentation of answers to questions with significant, logical errors;
- weak possession of the tools of the academic discipline, incompetence in solving standard (typical) tasks;
- inability to navigate in basic theories, concepts and directions on the discipline;

- work under the guidance of a teacher on practical, laboratory classes, the permissible level of the culture of performance of tasks.
- passivity on practical, laboratory classes, low level of the culture of performance of tasks.

2 (two) points, failed:

- fragmented knowledge within the educational standard of higher education;
- knowledge of individual literary sources, recommended by the curriculum of the institution of higher education on the discipline;
- inability to use scientific terminology of the academic discipline, the presence in the answer rude, logical errors;
- passivity on practical, laboratory classes, low level of the culture of performance of tasks.

1 (one) point, failed:

- lack of knowledge and (competences) within the educational standard of higher education, failure to answer, failure to appear for attestation without good cause.

11. METHODS AND MEANS OF IMPLEMENTATION OF THE CONTENT OF THE EDUCATIONAL PROGRAM AND TRAINING OF EDUCATIONAL, TRAINING AND METHODOLOGICAL MATERIALS

The training will be conducted using classical methods, project methods and distance learning technologies, implemented on the portal SDO.VSTU.BY. Trainees will be provided with access to the SDO.VSTU.BY portal.

In the classroom, students will learn the discipline directly in the computer class. The following software (software) will be used during the training:

- IDE MS Visual Studio Community,
- web-browser.

The following tools and technologies were used in the preparation of training, teaching and methodological materials:

- Adobe PDF.
- LMS Moodle.
- MS Visual Studio Community 2015.
- Programs for recording and editing video.

12. RESOURCES

Main literature

№	Authors	Bibliographic description
1.	Т. А. Павловская	С/С++. Программирование на языке высокого уровня: учебник для студентов вузов, обучающихся по направлению подготовки дипломированных специалистов "Информатика и вычислительная техника" . - Санкт-Петербург: Питер, 2010
2.	P.S. Deshpande	C & Data Structures = C и структура данных / P. S. Deshpande, O. G. Kakde. - New Delhi: Dreamtech Press, 2010.
3.	H. Schildt	С++: The Complete Reference = С++: полное руководство / H. Schildt,. - 4th ed. - New Delhi: Tata McGraw Hill Education Private Limited, 2010.
4.	В. В. Лаптев	С++. Объектно-ориентированное программирование:

.		учебное пособие - Санкт-Петербург: Питер, 2008. - 458 с
5.	Б. Керниган,, Д. Ригчи	Язык программирования Си: пер. с англ. /; под ред. В. С. Штармана. - 2-е изд., перераб. и доп. - Москва: Финансы и статистика, 1992.

Additional literature

6.	Страуструп Б.	Язык программирования C++, спец. изд. – М.,СПб.: “Издательство БИНОМ” – “Невский диалект”, 2001
7.	Дейтел Х, Дейтел П.	Как программировать на С. – М.: ЗАО “Издательство БИНОМ”, 2000.–1008с.
8.	Кнут Д.Э.	Искусство программирования: Учеб. пособие. Т. 3. Сортировка и поиск. – М.: Вильямс, 2000
9.	Д. Э. Кнут	Искусство программирования: пер. с англ. Т. 2 : Получисленные алгоритмы - 3-е изд. - Москва: Вильямс, 2001.
10.	Д. Э. Кнут	Искусство программирования: пер. с англ. Т. 1 : Основные алгоритмы - 3-е изд. - Москва: Вильямс, 2001.