Programming

	Prerequisites:	No prerequisites
	Learning outcomes:	As a result of discipline studying students should know: • the basic constructs, principles and mechanism of modern object-oriented language; be able: • to define and use classes; • to develop applications; • to use modern IDE . have an idea: • about memory volume and performance estimating.

	Lecture	**Hours**	**Laboratory works**		**Ref.**
			Content	**Hours**	
1	***Introduction.*** History of development of programming languages	1			
2	***Programming language C ++.*** A general characteristic of language C++. The language alphabet. Identifiers. Comments. The data and their types. Constants: whole, string. Variables. Arithmetic operations. Logic operations. Operations of relations. Mathematical library function.	2	Information Input-output on the screen.	4	[2],[3]
3	***Basic statements.*** Assignment operator. Rules of various types operands transformation. The operator "expression". Operators of increase and decrease. The empty operator. The compound operator.	2			[3],[4]
4	***Conditional statements.*** Full conditional operator **if-else.** The short operator **if**. The enclosed operators **if**.	1			
5	***Cycle statements.*** The **for** statement. The work scheme **for**. The enclosed cycles **for**. The **while** statement. The **while** scheme work. The operator of cycle **do-while**.	2	***Tabulation of functions*** Drawing up of programs by use of operators of a cycle **for**, **while, do-while.**	4	[2],[4]
6	***Arrays.*** Feature of array use. The description of array. The size of an array. Type of array. Section of arrays. Examples of one-dimensional and two-dimensional arrays use.	4	Drawing up of programs by use of one and two-dimensional arrays.	4	[3]
7	***Functions.*** Function definition. Type of returned value. Types and number of formal arguments. A function body. Prototype of function. Function call. Typical errors. Functions use examples.	4	Drawing up of programs by use of functions.	4	[2],[4]

8	***Strings.*** Definition of string. Internal representation of *string*. Two possibilities in processing of a *string* in CRT. Access to a class *string*. Operators of work with objects string type. Functions of work with strings. S*trings* use examples.	2	Drawing up of programs by use of standard string functions.	4	[2],[4]
9	***Classes and object.*** Class declaration Components of a Class. Class Data Member Functions. Constructors Destructors. Private Portion of a Class. Public Portion of a class Protected Portion of a Class Object. Simple Object Creation and Usage	2	create classes and objects in	4	[4], [6]
10	***Inheritance.*** Terms related to Inheritance. Base Class. Derived Class. Single Inheritance. Multiple Inheritance.	2	derive classes using inheritance	2	[4], [6]
11	***Polymorphism.*** Pointers to Derived Classes. Virtual Functions. Abstract Class. Dynamic Binding	2	writing polymorphic functions	2	[4], [6]
12	***Overloading.*** Function Overloading. Function Overloading With different number of arguments. Function Overloading. Operator Overloading. Overloading Binary Operators. Overloading Unary Operators.	2			[4], [6]
13	***Streams.*** Input Streams. Output Streams. File Streams. String Streams. Input/Output for user defined types.	3	working with streams	2	[4], [6]
14	***The conclusion.*** Modern technologies of programming.	1			[1]
	Total	30		30	

The *basic:*

1. Vjukova N.I., Galatenko V. A, Hodulev A.B. Regular approach to programming. M: the Science. 1988.

2. E.A.Ishkova. The beginnings of C and C ++ programming. Moscow: 2000

3. Stephen Prata. The C language programming. Kiev: 2000.

4. H. M.Dejtel. How to program on C ++. M: Binom 2000.

The additional:

5. Jeff Eldzher. C++. Library of the programmer. Publishing house: Peter, 1999

6. B.Straustrup. Design and evolution of language C ++. M: the Press, 2000