FACULTY OF COMPUTER SCIENCE AND ENGINEERING

CURRICULUM

Confirmed:
Dean:

CS203

OBJECT-ORIENTED PROGRAMMING

Updated: protocol № 7/30.01.2019

Lecturer: Assoc.Prof. Veselina Zhecheva, PhD

ANNOTATION

The course provides knowledge about the principles of working with objects and the object-oriented approach in software development. Using the C++ programming language, skills are formed to work with classes, constructors, destructors, mutators, access functions, static and dynamic data structures, static and dynamic memory allocation and management, stream handlers, templates and data inheritance.

BASIC PURPOSES

The main objective of the course is to get familiar to the principles of object-oriented programming using the C++ language. Upon successful completion of the Object Oriented Programming course, students will:

• define and use record-type structures and know the mechanisms for dynamic memory allocation and organization of dynamic data structures;

• know the basic elements of OOP: classes, data encapsulation, member functions, constructors and destructors, object constants and constant class members,

• model the subject area by inheriting classes and redefining operations;

• know and use virtual functions and polymorphism, input and output streams, and patterns.

PREREQUISITES

Students should have knowledge of structured programming with C++ from the Programming course.

STATUS AND STRUCTURE

PROGRAMME	status	ECTS	Full time				Part time			
			л	с	у	общ	л	с	у	Общ
Computer Science	Mandatory	7	30	40		70	15	20		35
Software Engineering	Mandatory	7	30	40		70	15	20		35

COURSE CONTENT

Topic 1. Basic composite data structures. Array type. Access the array elements with indexed variables. Structure type (record). Memory alocation. Elementary constructors and default constructors. Access to structure members. Encapsulation of information. Declaration of members. Scope of declarations.

Topic 2. Pointers and references. & and * operators. Access data by name and address. Pointers and arrays. Pointers and Strings. Static and dynamic memory. NEW and DELETE operators.

Topic 3. Dynamic data structures. Organization and memory allocation of data. Access to data. Linear dynamic structures. Lists - single and double linked. List operations. Stack and queue. Management strategies. Allocating and releasing memory. Applications.

Topic 4. Non-linear dynamic structures. Trees - definition, representation methods and applications. Binary trees. Representation and Applications. Ordered binary trees and operations. Graphs.

Topic 5. Classes. Definition. Member variables (properties) and functions (methods). Memory access and placement. Objects and scope. Data encapsulation. Separating the interface from the class implementation.

Topic 6. Constructors and destructors. Access methods and auxiliary methods. Mutators. Constant objects and member functions. Composition of classes. Functions inline**.**

Topic 7. Objects. This pointer. Cascading calls to member functions. Friendly functions and classes. Static class members. Proxy classes**.**

Topic 8. Class Inheritance. Base and derived classes. Types of inheritance. Access to members. Overriding base class members in derived classes. Composition and Inheritance.

Topic 9. Virtual functions and polymorphism. Abstract classes. Dynamic and static binding. Operations overloading.

Topic 10. Templates. Function and class templates. Create groups of related types. Overloading template functions.

Topic 11. Exceptions. Handling errors and other exceptional situations. Throwing and catching exceptions. Synchronous and asynchronous events. Regeneration and specification of exceptions. Exceptions and resource management.

Topic 12. Streams. Stream input and output. Input and output formatting. Hierarchy of input-output operations. User input and output handlers. Input/output of user variables. Success and failure of I/O operations.

Topic 13. Standard templates library (STL). Containers, iterators, algorithms, functors. Stacked and unstacked containers.

SEMINARS

Topic 1. Basic Composite Data Structures. Array type.
Topic 2. Basic Composite Data Structures. Structure type.
Topic 3. Simple classes. Objects and functions.
Topic 4. Pointers and references.
Topic 5. Scope of class declarations. INLINE functions.
Topic 6. Objects. This pointer.
Topic 7. Derived classes. Inheritance and access. Overloading.
Topic 8. Constructors, operator functions and destructors of derived classes. Inheritance and templates.
Topic 9. Linear dynamic structures. Queue.
Topic 10. Linear dynamic structures. Stack. Filling and emptying a stack. NEW and DELETE operators.
Topic 12. Nonlinear dynamic structures. A binary tree. Binary Tree Algorithms.
Topic 13. Streams. Working with input and output.
Topic 14. Standard templates library (STL). Applications.

PLANNED LEARNING ACTIVITIES AND TEACHING METHODS

Training methods:

Face-to-face lectures and seminars
Visual learning
Practical Education
Interactive learning
E-learning through the Moodle platform

Teaching tools:

Self-paced work
Educational video materials incl. video presentations

Practical tasks

Programming tasks using application software
Use of electronic resources in the Moodle platform: theoretical materials, presentations, sample programs, tests and tasks for self-paced work on each topic

COURSEWORK

Each student is assigned two individual tasks, which are developed during the semester: respectively on single classes and on base and derivative classes. The presentation takes place no later than the end of the semester.

ASSESSMENT METHODS

Each student's work during the semester is evaluated with a grade. It is obtained as a result of the development of two courseworks - on single classes and on base and derivative classes, one control work, five homeworks, as well as from the student's participation in lectures and exercises.
The final exam is a test, containing both open-ended and closed-ended questions. The final grade is formed as a sum of the semester exam and the current grade.

To achieve the final assessment, the student collects points with the maximum value 100. The distribution of points by evaluated activities is as follows:

1. Face-to-face assessment..... .... 16 points

1.1. Control work - 10 points

1.2. Seminars - 6 points

(visits, participation in seminars)

2. Outside classroom employment .... 30 points

2.1. Course assignments – 2 pcs. - 20 points

2.2. Homework - 10 points

3. Final exam……..... .... 54 points

The student must have a minimum of 14 points from the control during the semester and a minimum of 22 points from the examination procedure in order to form a comprehensive assessment. The final grade is formed by distributing the points on the scale:

• from 54 to 60 points - Medium (3);

• from 61 to 70 points - Good (4);

• from 71 to 80 points - Very good (5);

• from 81 to 100 points - Excellent (6).