OBJECT ORIENTED PROGRAMMING

COURSE OUTLINE

Responsible: Marios Antonakakis

1. GENERAL			
SCHOOL	School of Engineering		
ACADEMIC UNIT	Department of Electronic Engineering		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE	0806.2.002.0	**SEMESTER**	1st
COURSE TITLE	Object Oriented Programming		

INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course	**WEEKLY TEACHING HOURS**	**CREDITS**
	4	5
Total	**4**	**5**

COURSE TYPE general background, special background, specialised general knowledge, skills development	General Background, Skills Development
PREREQUISITE COURSES	Structured Programming
LANGUAGE OF INSTRUCTION and EXAMINATIONS	English
OFFERED TO ERASMUS STUDENTS	Yes (in English) — Spring Semester
COURSE WEBSITE (URL)	https://eclass.hmu.gr/courses/EE335/

2. LEARNING OUTCOMES

Learning outcomes

Upon successful completion of the course, students will be able to:

- Analyse and solve problems using object-oriented thinking.
- Design and implement object-oriented programs in C++.
- Manage memory and resources effectively.
- Apply fundamental software design principles (OOP).
- Develop and compile programs in a Unix environment.
- Document software design using UML class diagrams.

General Competences

Search, analysis and synthesis of data and information, using the necessary technologies

Adaptation to new situations

Decision-making

Autonomous work

Teamwork

Project planning and management

Exercise of criticism and self-criticism

Promotion of free, creative and inductive thinking

3. SYLLABUS

The course “Object-Oriented Programming” provides a systematic introduction to the principles, techniques and methodology of object-oriented programming, using C++ as the main implementation language and the Unified Modeling Language (UML) for model design. The aim is to develop object-oriented thinking and an understanding of how modern software applications are designed, organised and implemented. The course covers the fundamental concepts of classes and objects, data access levels, constructors and destructors, the copy constructor and assignment operator, as well as resource and memory management through pointers and dynamic allocation. Particular emphasis is placed on the correct use of references, constant expressions (const, constexpr), operator overloading and string handling. Fundamental object-oriented relationships such as composition, aggregation, inheritance and polymorphism are presented and analysed, together with the use of virtual functions. Generic programming through templates and the implementation of basic data structures, including arrays, vectors of objects and linked lists, are also examined. An introduction to exception handling and recursion is provided. Students become familiar with the Unix development environment and the compilation and linking process for C++ programs, while also being introduced to UML class diagrams for the analysis and documentation of object-oriented systems. Teaching combines theoretical instruction with laboratory practice, aiming to develop skills in designing, implementing and evaluating software that follows the object-oriented paradigm and modern development best practices.

Lectures to be delivered:

1. Fundamental introductory concepts in object-oriented programming
2. Access control and constructors
3. Unix / Docker environment and compilation of C++ files
4. The friend keyword, conversion constructors, destructors
5. The const and constexpr keywords, pass by value and pass by reference
6. Operator overloading, composition and aggregation
7. Arrays of objects and vectors of objects
8. Pointers, dynamic memory allocation and recursion
9. Copy constructor and copy assignment
10. C-Strings, std::string, std::string_view and the [] and & operators
11. Inheritance, polymorphism and virtual functions
12. Templates, linked lists and exceptions
13. Useful C++ concepts, basic data structures, and C++ vs. Java
14. UML class diagrams

4. TEACHING and LEARNING METHODS - EVALUATION

DELIVERY Face-to-face, Distance learning, etc.	Face-to-face theoretical teaching. Assignments in small student groups. Laboratory exercises on object-oriented programming applications in small student groups. Computer-based laboratory training using appropriate software.

USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY Use of ICT in teaching, laboratory education, communication with students	Use of presentation software. Communication with students through an asynchronous e-learning platform. Use of g++, VS Code and CodeBlocks for developing software applications using object-oriented programming. Use of C++ and UML.
TEACHING METHODS The manner and methods of teaching are described in detail.	**Activity** / **Semester workload** Lectures: 26 Application Exercises in Small Groups: 13 Laboratory Assigments: 13 Invidual Application Exercise Assigments: 39 Individual Study: 42 Exams: 4 **Course total: 137**
STUDENT PERFORMANCE EVALUATION Description of the evaluation procedure	I. Written final examination (FE) (50%) - Problem solving/calculations - Comparative evaluation of theoretical concepts II. Semester assignment (SA) (20%) - Laboratory assignments/technical reports/measurements in small groups III. Laboratory examination (LE) (30%) - Laboratory assignments/technical reports/measurements in small groups The course grade (FE*0.5 + LE*0.3 + 0.2*SA) must be at least five (5). The grade for each of I, II and III must be at least three (5). The assessment criteria are available to students on the course website and are announced during the first class.

5. ATTACHED BIBLIOGRAPHY

- *Recommended Bibliography:*

Greek or translated textbooks:

- Object-Oriented Programming in C++, Robert Lafore, 1st edition: 09/02/2006, Pages: 1040, ISBN: 960-209-904-6, Code: 33029, Eudoxus Code: 13598.
- C++ in Depth – 2nd edition, M. Chatzigiannakis, Kleidarithmos Publications, ISBN: 978-960-461-620-6
- C++ Programming – 6th edition, P. Deitel and H. Deitel, Giourdas Publications, ISBN: 9605125919

Foreign-language textbooks:

- Programming Principles and Practice Using C++, Bjarne Stroustrup, ISBN: 9780138308681
- Beginning C++ Through Game Programming, M. Dawson, ISBN: 978-1-305-10991-9

C++ How to Program, P. Deitel and H. Deitel, ISBN: 9780134448237