

COURSE OUTLINE

(1) GENERAL

SCHOOL	SCHOOL OF ENGINEERING		
ACADEMIC UNIT	Department of Informatics and Computer Engineering		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE	ICE-4004	SEMESTER	4th
COURSE TITLE	APPLICATION DEVELOPMENT METHODOLOGIES		
INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course, e.g. lectures, laboratory exercises, etc. If the credits are awarded for the whole of the course, give the weekly teaching hours and the total credits		WEEKLY TEACHING HOURS	CREDITS
Lectures		3	
Laboratories		2	
		5	5
Add rows if necessary. The organisation of teaching and the teaching methods used are described in detail at (d).			
COURSE TYPE general background, special background, specialised general knowledge, skills development	Special Background		
PREREQUISITE COURSES:			
LANGUAGE OF INSTRUCTION and EXAMINATIONS:	Greek		
IS THE COURSE OFFERED TO ERASMUS STUDENTS	No		
COURSE WEBSITE (URL)	https://eclass.uniwa.gr/courses/CS145/		

(2) LEARNING OUTCOMES

Learning outcomes

The course learning outcomes, specific knowledge, skills and competences of an appropriate level, which the students will acquire with the successful completion of the course are described. Consult Appendix A

- Description of the level of learning outcomes for each qualifications cycle, according to the Qualifications Framework of the European Higher Education Area
- Descriptors for Levels 6, 7 & 8 of the European Qualifications Framework for Lifelong Learning and Appendix B
- Guidelines for writing Learning Outcomes

This course considers the various programming methodologies and it explains the reasons that each methodology is introduced. We consider successively various Programming paradigms: Structured programming, Functional Programming, Logic programming, Object Oriented Programming, Component based Programming and Rapid application development (RAD). With this course, students acquire a deep knowledge of the object programming model. They also learn how to elaborate components within rapid application development environments. They also become familiar with the rapid development of Interfaces and especially of Graphical User Interfaces. The mentioned knowledge, students will be able to use for designing, composing and creating integrated applications. We also investigate the most widespread Architectural patterns, such as: Model View Controller, Hierarchical Model View Controller, Model View Adapter, Observer pattern.

In particular, the student who successfully concludes the course:

- Utilizes the concept of Abstract data types.
- Classifies programming languages.
- Design and Implement Object-Oriented Programmes.

- Implement Applications based on Events.
- Develop Applications that utilize Components (Component-Oriented Programming).
- Design Interfaces through Rapid application development Environments.
- Implement Interfaces by using only Components and Handle the events (build Event Handlers) manually.
- Compares the various RAD (Rapid application development) Environments.
- Implement applications with the Object Oriented Languages (e.g. Java) but also implement applications with the RAD Environments (e.g. NetBeans, Eclipse, IntelliJ, VB.NET, C++ Builder, Visual Studio, etc).
- Implement Multi-Threading applications

General Competences

Taking into consideration the general competences that the degree-holder must acquire (as these appear in the Diploma Supplement and appear below), at which of the following does the course aim?

Search for, analysis and synthesis of data and information, with the use of the necessary technology	Project planning and management
Adapting to new situations	Respect for difference and multiculturalism
Decision-making	Respect for the natural environment
Working independently	Showing social, professional and ethical responsibility and sensitivity to gender issues
Team work	Criticism and self-criticism
Working in an international environment	Production of free, creative and inductive thinking
Working in an interdisciplinary environment
Production of new research ideas	Others...

- Search for, analysis and synthesis of data and information, with the use of the necessary technology
- Working independently
- Team work
- Working in an interdisciplinary environment
- Production of free, creative and inductive thinking
- Adapting to new situations

(3) SYLLABUS

- Classes and Objects
- Console Applications
- Single Tier Applications with Graphical User Interfaces
- Applications running into a Browser's Environment
- Dialog items
- Layouts for Dialog Items
- Events and event handlers
- Programmatically design of Graphical Interfaces
- Advanced Software Constituents (Components)
- Exceptions and Exception Handling
- Multi-Thread Applications
- Hierarchy of Classes for Input and Output
- Data Structures (Tables, Lists, Trees, Maps)
- Interfacing with Databases

- Utility Based Graphical User Interface design
- Case study implementation for a Complex Application

(4) TEACHING and LEARNING METHODS - EVALUATION

DELIVERY Face-to-face, Distance learning, etc.	Face-to-face	
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY Use of ICT in teaching, laboratory education, communication with students	Java, NetBeans, Eclipse, MySQL, PostgreSQL, Support of Learning process with the Asynchronous Tele-Education platform Open eClass	
TEACHING METHODS The manner and methods of teaching are described in detail. Lectures, seminars, laboratory practice, fieldwork, study and analysis of bibliography, tutorials, placements, clinical practice, art workshop, interactive teaching, educational visits, project, essay writing, artistic creativity, etc. The student's study hours for each learning activity are given as well as the hours of non-directed study according to the principles of the ECTS	Activity	Semester workload
	Lectures	39
	Laboratory practice	26
	Projects for implementing console applications, GUI applications and Applications running into a Browser's Environment	30
	Independent personal study	30
	Course total	125
STUDENT PERFORMANCE EVALUATION Description of the evaluation procedure Language of evaluation, methods of evaluation, summative or conclusive, multiple choice questionnaires, short-answer questions, open-ended questions, problem solving, written work, essay/report, oral examination, public presentation, laboratory work, clinical examination of patient, art interpretation, other Specifically-defined evaluation criteria are given, and if and where they are accessible to students.	<p>I. Written final exams (60%) that includes:</p> <ul style="list-style-type: none"> - multiple choice questions - short-answer questions - problem solving (implementations of Console applications, Graphical User Interfaces applications, applications for the Browser's Environment) <p>II. Laboratory work and oral examination (for implementing Console applications, Graphical User Interface applications, applications for the Browser's Environment) (40%)</p>	

(5) ATTACHED BIBLIOGRAPHY

- Suggested bibliography:

1. Νικήτα Ν. Καρανικόλα, Java για όλους, Εκδόσεις Νέων Τεχνολογιών, 2015.
2. Κυτάγιας Χρήστος, Κυτάγιας Κώστας, Κυτάγιας Δημήτριος, Πρεζεράκος Γεώργιος, Αντικειμενοστρεφής Προγραμματισμός με Java, Σύγχρονη Εκδοτική Ε.Π.Ε., 2013.
3. Λιακέας Γ., Εισαγωγή στην Java 2, Κλειδάριθμος, 2003.
4. Herbert Schildt, Java 2, The Complete Reference, Fifth Edition, 2002, McGraw Hill/Osborne,
5. Laura Lemay και Rogers Cadenhead, Πλήρες Εγχειρίδιο της Java 2, Ελληνική Έκδοση, Εκδότης Μ. Γκιούρδας,
6. Holzner S., Teach Yourself Visual Basic.NET in 21 Days, (Sams Teach Yourself), Sams Publishing, 2003.
7. Szyperski C., Component Software, Addison-Wesley, 1998.

8. Spencer K., Eberhard T., Alexander J., *OOP: Building Reusable Components with Microsoft® Visual Basic® .NET*, Microsoft Press, 2002.
9. Dobson R., *Programming Microsoft® Visual Basic® .NET for Microsoft Access Databases*, Microsoft Press, 2003.

- Related academic journals: