Syllabus

1. Program information

1.1. Institution	ACADEMY OF ECONOMIC STUDIES
1.2. Faculty	Economic Cybernetics, Statistics and Informatics
1.3. Departments	(Departament) INFORMATICA SI CIBERNETICA ECONOMICA
1.4. Field of study	Economic Informatics
1.5. Cycle studies	Master Studies
1.6. Education type	Full-time
1.7. Study program	IT&C Security
1.8. Language study	English
1.9. Academic year	2016-2017

2. Course information

2.1. Name	**Secure Applications Programming**								
2.2. Code	**16.0241IF1.1-0005**								
2.3. Year of studies	**1**	2.4. Semester	**1**	2.5. Assessment type	**Exam**	2.6. Course type	**O**	2.7. Number of ECTS	**5**
2.8. Instructors									

3. Total estimated time

3.1. Number of weeks	14.00		
3.2. Number of hours per week	5.00	of which	
		C(C)	3.00
		S(S)	2.00
3.3. Total hours from curriculum	70.00	of which	
		C(C)	42.00
		S(S)	28.00
3.4. Total hours of study per semester (ECTS*25)	125.00		
3.5. Total hours of individual study	55.00		

Time distribution for individual study	
Study the textbook, course support, bibliography and notes	
Further reading in the library, on the online platforms and field	
Preparing seminars, labs, homework, portfolios and essays	
Tutoring	
Examinations	
Other activities	

4. Prerequisites

4.1. About curriculum	Cryptography Basis
4.2. About skills	Ways to organize data in computer memory, models and requirements of data usage in the source program, use and allocation of memory areas; Knowledge of syntax elements of programming languages C + + and Java and how to build, compile, execute and debug of the program source, usinf integrated development environments for software applications; Knowledge of to implement specific elements of object oriented programming languages C + + and Java.

5. Requirements

C(C)	Lectures held in rooms with multimedia teaching equipment (video-projector).
S(S)	Rooms with computers installed with programming environments C + + and Java, and Ubuntu within virtual machines with GCC, Java plus necessary tools.

6. Skills covered

	C3	Using modern computer technologies for developing components that ensure maximum IT security
	CT3	Taking training need to create prerequisites for progress in career and adapt the professional and managerial skills to the dynamic of the economic environment

7. Course objectives

7.1. General objective	Standard knowledge of C++ and Java programming languages for implementing cryptographic applications.
7.2. Specific objectives	Cunoașterea modului de implementare a elementelor criptografice în limbajele de programare C++ şi Java, utilizând biblioteci criptografice; Dezvoltarea aplicaţiilor criptografice elaborate în C++ şi Java; Dezvoltarea aplicaţiilor criptografic folosind interfeţe grafice cu utilizatorul.

8. Course contents

8.1. C(C)		Teaching methods	Advices
1	Review: data types, variables pointer, use stack and heap memory in C++.	PowerPoint support, examples of implementation in development environment, interaction with students.	Electronic media distribution , documentary sources recommending, implementing exercises.
2	Functions: mechanisms for parameter transferring, pointers to functions in C++.	PowerPoint support, examples of implementation in development environment, interaction with students.	Electronic media distribution , documentary sources recommending, implementing exercises.
3	Files: types, access to data stored on disk, processing data stored in files in C++.	PowerPoint support, examples of implementation in development environment, interaction with students.	Electronic media distribution , documentary sources recommending, implementing exercises.
4	Classes: defining, structuring, construction and destruction of objects, the pointer this in C++.	PowerPoint support, examples of implementation in development environment, interaction with students.	Electronic media distribution , documentary sources recommending, implementing exercises.

5	Classes: copy constructor and operator overloading in C++.	PowerPoint support, examples of implementation in development environment, interaction with students.	Electronic media distribution , documentary sources recommending, implementing exercises.
6	Classes: derivation and polymorphism in C++.	PowerPoint support, examples of implementation in development environment, interaction with students.	Electronic media distribution , documentary sources recommending, implementing exercises.
7	Classes: building class template and use Standard Template Library in C++.	PowerPoint support, examples of implementation in development environment, interaction with students.	Electronic media distribution , documentary sources recommending, implementing exercises.
8	Build and use static and dynamic libraries in C++.	PowerPoint support, examples of implementation in development environment, interaction with students.	Electronic media distribution , documentary sources recommending, implementing exercises.
9	OpenSSL: use in cryptographic applications in C++.	PowerPoint support, examples of implementation in development environment, interaction with students.	Electronic media distribution , documentary sources recommending, implementing exercises.
10	OpenSSL: structuring and implementing digital certificate in C++ using OpenSSL.	PowerPoint support, examples of implementation in development environment, interaction with students.	Electronic media distribution , documentary sources recommending, implementing exercises.
11	Developing cryptographic applications using OpenSSL with MFC GUI.	PowerPoint support, examples of implementation in development environment, interaction with students.	Electronic media distribution , documentary sources recommending, implementing exercises.
12	Basic concepts of Java: JDK, JVM, JRE. Use Eclipse / Netbeans to write, test and run Java programs.	PowerPoint support, examples of implementation in development environment, interaction with students.	Electronic media distribution , documentary sources recommending, implementing exercises.
13	Concepts of object-oriented programming. Java vs. C + +.	PowerPoint support, examples of implementation in development environment, interaction with students.	Electronic media distribution , documentary sources recommending, implementing exercises.
14	Features of Java. String, immutable, wrapper classes.	PowerPoint support, examples of implementation in development environment, interaction with students.	Electronic media distribution , documentary sources recommending, implementing exercises.

15	Advanced Java topics: serialization, Java generics, add-notation, threads.	PowerPoint support, examples of implementation in development environment, interaction with students.	Electronic media distribution , documentary sources recommending, implementing exercises.
16	Java collection framework	PowerPoint support, examples of implementation in development environment, interaction with students.	Electronic media distribution , documentary sources recommending, implementing exercises.
17	I/O streams files and network	PowerPoint support, examples of implementation in development environment, interaction with students.	Electronic media distribution , documentary sources recommending, implementing exercises.
18	Java Native Interface	PowerPoint support, examples of implementation in development environment, interaction with students.	Electronic media distribution , documentary sources recommending, implementing exercises.
19	Design patterns: Factory and singleton	PowerPoint support, examples of implementation in development environment, interaction with students.	Electronic media distribution , documentary sources recommending, implementing exercises.
20	Java Crytpography Architecture: hash functions, symmetric algorithms (DES / AES) in ECB and CBC mode.	PowerPoint support, examples of implementation in development environment, interaction with students.	Electronic media distribution , documentary sources recommending, implementing exercises.
21	Java Crytpography Architecture: public key systems (RSA) and digital certificates	PowerPoint support, examples of implementation in development environment, interaction with students.	Electronic media distribution , documentary sources recommending, implementing exercises.
22	Developing cryptographic applications using Java Swing	PowerPoint support, examples of implementation in development environment, interaction with students.	Electronic media distribution , documentary sources recommending, implementing exercises.

Bibliography

- Ion Smeureanu, Marian Dardala, Programarea orientata obiect in limbajul C++, CISON, 2002, România
- Ion Smeureanu, Programarea in limbajul C/C++, CISON, 2001, România
- Irina Athanasiu , Programarea practica in limbajul Java, ALL, 2001, România
- Ion Ivan, Cristian Toma , Informatics Security Handbook - 2nd Edition, ASE Publishing House, 2009, România
- David Hook, Beginning Cryptography in Java, Wrox, Statele Unite ale Americii
- Jonathan Knudsen, Patrick Niemeyer, Learning Java, 3rd Edition, O'Reilly, Statele Unite ale Americii

8.2. S(S)	Teaching methods	Advices

1	Review: examples of implementation in C++ of data types, pointer variables, reservation and allocation stack memory heap.	Explaining the mechanisms used at compile-time and running-time for an software application.	Using of computer installed with programming environment.
2	Examples of implementation in C++ of functions: explaining of transfer mechanisms of the parameters, pointers to functions.	Explaining the mechanisms used at compile-time and running-time for an software application.	Using of computer installed with programming environment.
3	Examples of implementation in C++ of files: highlight content types and organization on certain standard formats, access to data stored on disk, processing data stored in files.	Explaining the mechanisms used at compile-time and running-time for an software application.	Using of computer installed with programming environment.
4	Examples of implementation in C++ of classes: definition, types of attributes and methods, and access to, building and destroying objects, the pointer this .	Explaining the mechanisms used at compile-time and running-time for an software application.	Using of computer installed with programming environment.
5	Examples of implementation in C++ of classes: definition of copy constructor and its cases of calling, overloading of operators.	Explaining the mechanisms used at compile-time and running-time for an software application.	Using of computer installed with programming environment.
6	Examples of implementation in C++ of classes: implementing polymorphism through virtual functions.	Explaining the mechanisms used at compile-time and running-time for an software application.	Using of computer installed with programming environment.
7	Examples of implementation in C++ of classes: construction of template classes and use of Standard Template Library in C++.	Explaining the mechanisms used at compile-time and running-time for an software application.	Using of computer installed with programming environment.
8	Examples of implementation in C++ for building and using static libraries (LIB) and dynamic (DLL) in command line and development environment.	Explaining the mechanisms used at compile-time and running-time for an software application.	Using of computer installed with programming environment.
9	Examples of implementation in C++ of cryptographic applications using OpenSSL library.	Explaining the mechanisms used at compile-time and running-time for an software application.	Using of computer installed with programming environment.
10	Structure and implementation of a digital certificate in C + + using OpenSSL library.	Explaining the mechanisms used at compile-time and running-time for an software application.	Using of computer installed with programming environment.
11	Developing MFC GUI cryptographic applications using OpenSSL.	Explaining the mechanisms used at compile-time and running-time for an software application.	Using of computer installed with programming environment.
12	Using NetBeans / Eclipse to develop Java applications	Explaining the mechanisms used at compile-time and running-time for an software application.	Testing presented topics by writing Java programs. Analysis of examples provided as laboratory support.

13	Java examples of implementing object-oriented programming concepts.	Explaining the mechanisms used at compile-time and running-time for an software application.	Testing presented topics by writing Java programs. Analysis of examples provided as laboratory support.
14	Java examples of object serialization.	Explaining the mechanisms used at compile-time and running-time for an software application.	Testing presented topics by writing Java programs. Analysis of examples provided as laboratory support.
15	Java examples of generics and Java Collection Framework	Explaining the mechanisms used at compile-time and running-time for an software application.	Testing presented topics by writing Java programs. Analysis of examples provided as laboratory support.
16	Java examples of threads	Explaining the mechanisms used at compile-time and running-time for an software application.	Testing presented topics by writing Java programs. Analysis of examples provided as laboratory support.
17	Java examples of I/O streams, files and neworking.	Explaining the mechanisms used at compile-time and running-time for an software application.	Testing presented topics by writing Java programs. Analysis of examples provided as laboratory support.
18	Java examples of cryptographic solutions: using Java Cryptography Architecture and Bouncy Castle	Explaining the mechanisms used at compile-time and running-time for an software application.	Testing presented topics by writing Java programs. Analysis of examples provided as laboratory support.
19	Management of digital certificates in Java	Explaining the mechanisms used at compile-time and running-time for an software application.	Testing presented topics by writing Java programs. Analysis of examples provided as laboratory support.
20	Cryptographic application development with Java Swing GUI.	Explaining the mechanisms used at compile-time and running-time for an software application.	Testing presented topics by writing Java programs. Analysis of examples provided as laboratory support.

Bibliography

- Ion Smeureanu, Marian Dardala, Programarea orientata obiect in limbajul C++, CISON, 2002, România
- Ion Smeureanu, Programarea in limbajul C/C++, CISON, 2001, România
- Irina Athanasiu , Programarea practica in limbajul Java, ALL, 2001, România
- Ion Ivan, Cristian Toma , Informatics Security Handbook - 2nd Edition, ASE Publishing House, 2009, România
- David Hook, Beginning Cryptography in Java, Wrox, Statele Unite ale Americii
- Jonathan Knudsen, Patrick Niemeyer, Learning Java, 3rd Edition, O'Reilly, Statele Unite ale Americii

9. Course contents corroboration with the demands of epistemic community representatives, professional associations and representative employers

In the software industry, products have a very high level of complexity determined by the number of components, linkages, deploying algorithms, implemented functionality, software nature of the problem that is addressed and their geographical distribution.

In the software engineering knowledge of cryptographic techniques, their implementation in various programming languages and classes of operations are implemented based on their complexity is essential in improving computer security software and characterize it. These elements are key factors in achieving high quality software products made with minimal cost and safe operation of their process.

Taking into account the best practices from ICT field applied by big companies such as: Intel, Oracle, Microsoft, IBM, HP and professional consortiums such as: Apache, Red Hat, ISO/IEC.

10. Assessment

Activity	Assessment criteria	Assessment methods	Percentage in the final grade
10.1. C(C)	Questions and comments on examples and exercises for implementation	Interaction consistency in course work.	10.00
10.2. S(S)	Practical tests.	Implementation way in programming language of cryptographic methods and responses to the teacher.	15.00
10.3. S(S)	Applied activities.	Techniques and methods of implementation of cryptographic components in applications developed in C + + and Java.	15.00
10.4. Final assessment	Knowledge gained during the course.	The evaluation of programming language implementation of cryptographic methods and responses to the teacher.	60.00

10.5. Grading scale	Whole notes 1-10
10.6. Minimum performance standard	Knowledge required: reservation / allocation and stack and heap memory management, object oriented programming concepts implemented in programming language (class, encapsulation, constructors, destructors, inheritance, polymorphism, class templates), static and dynamic libraries, installation and use OpenSSL applications with graphical user interface. The point granted by default is included in the weights assigned to the types of assessments.

Completion date,
07/10/2016

Instructors,

Approval date of department

Director of department,