UNIVERSITÀ DEGLI STUDI DI PALERMO

DEPARTMENT	Ingegneria
ACADEMIC YEAR	2017/2018
BACHELOR'S DEGREE (BSC)	COMPUTER ENGINEERING
SUBJECT	PROGRAMMING
TYPE OF EDUCATIONAL ACTIVITY	B
AMBIT	50289-Ingegneria informatica
CODE	05871
SCIENTIFIC SECTOR(S)	ING-INF/05
HEAD PROFESSOR(S)	LA CASCIA MARCO Professore Ordinario Univ. di PALERMO
OTHER PROFESSOR(S)	
CREDITS	9
INDIVIDUAL STUDY (Hrs)	144
COURSE ACTIVITY (Hrs)	81
PROPAEDEUTICAL SUBJECTS	
MUTUALIZATION	
YEAR	2
TERM (SEMESTER)	1° semester
ATTENDANCE	Not mandatory
EVALUATION	Out of 30
TEACHER OFFICE HOURS	**LA CASCIA MARCO** Monday 15:00 17:00 Microsoft Teams Codice: wztkv0u

DOCENTE: Prof. MARCO LA CASCIA

PREREQUISITES	Basic knowledge of algorithms and data structure. Knowledge of data representation in a computer. Basic C programming skills.
LEARNING OUTCOMES	- Knowledge and understanding The student will acquire knowledge about object oriented programming and design and development and maintenance of Java programs. In particular the student will know Java language features such as operators, functions, control flow, classes and objects, inheritance, polymorphism, interfaces, generics, exception handling, collection, I/O. To achieve this goal, the course will include: lectures; class discussions of sample code. - Applying knowledge and understanding The student will acquire the practical capabilities to design and code software using Java advanced features and will be able to maintain and update existing programs. To achieve this goal, the course will include sessions in the computer lab writing Java programs. - Making judgements The student will acquire the capabilities needed to analyze third part classes and libraries and consider their possible use in developing complex software. To achieve this goal the course will include: analysis and discussion of case studies; analysis of the Java standard library with focus on advantages and disadvantages of its use. - Communication skills The student will acquire the skills needed to discuss problems related to the course topics. To achieve this goal the course include computer lab session where students explain how they solve the assignments and the difficulties encountered. - Learning skills The student will be able to solve autonomously any problem related to object oriented software development and will be able to deepen his knowledge on complex topics such as polymorphism, dynamic memory management, concurrency, software scalability, etc... To achieve this goal the course include: exercises to solve autonomously; discussion on difficulties encountered.
ASSESSMENT METHODS	Assessment methods will focus on the evaluation of learning outcomes of the course (see below) according to the Dublin descriptors. The final grade will be from 18/30 to 30/30 cum laude. - Evaluation of knowledge and understanding This objective will be assessed by an oral discussion concerning the topics of the syllabus. This objective will count as 30% of the final grade. - Evaluation of applying knowledge and understanding This objective will be assessed by developing a complete Java program based on the textual description of the problem to solve. This objective will count as 35% of the final grade. - Evaluation of making judgements This objective will be assessed by developing a complete Java program. In developing the program the student has to perform design choices autonomously. This objective will count as 15% of the final grade. - Evaluation of communication skills This objective will be assessed by the oral discussion concerning the topics of the syllabus and the developed Java program. This objective will count as 10% of the final grade. - Evaluation of learning skills This objective will be assessed by means of the discussion of topics introduced in class but requiring a significant independent study from the student. This objective will count as 5% of the final grade.
EDUCATIONAL OBJECTIVES	The course present in detail object oriented programming in Java. General object-oriented programming techniques will be studied together with specific Java programming aspects such as packages, javadoc, concurrent programming and Java library. The most commonly used data structure will also be studied with reference to their implementation in the Java library.
TEACHING METHODS	Lectures, computer laboratory sections
SUGGESTED BIBLIOGRAPHY	CLAUDIO DE SIO CESARI (2014). Manuale di Java 8. Programmazione orientata agli oggetti con Java standard edition 8. Hoepli

	PELLEGRINO PRINCIPE (2014). Java 8. Apogeo. PATRICK NIEMEYER. Learning Java 4th edition. O'Really

SYLLABUS

Hrs	Frontal teaching
2	Introduction to Java programming language
2	Parts of a Java program: object-oriented fundamentals, methods, variables, constructors, packages.
4	Identifiers, data types, array. Primitive data types.
4	Operators and execution flow of a program: basic operators, manipulating the control flow.
6	Encapsulation and visibility: object-oriented programming paradigm, encapsulation, access modifiers, static modifier.
4	Inheritance and interfaces: inheritance, final modifier, Object class, inheritance and encapsulation, interfaces.
4	Polimorphism: method polimorphism, overload and override, data polimorphism.
4	Exceptions and assertions: exception handling in Java, exception propagation, introduction to assertions.
2	Enumeration and nested types: inner classes, anonymous classese, enumeration types.
2	Generic types: creating generic types, type inference.
4	Java library: String, Object, System, Runtime and Math classes, java.util package, Date-Time API.
2	Thread management: Thread class, thread syncronization, communication between threads, concurrency.
4	Collections: Collection interface, List, Queue and Deque, Map e SortedMap, algorithms
2	Input/Output: character and byte stream, keyboard input, file management, objects serialization.
2	Graphical interfaces: JavaFX, creating interfaces with Layout, event handling.

Hrs	Practice
6	Java Development Environment. Simple Java programs.
3	Implementation of simple algorithm in Java.
3	Implementation of programs using array.
6	Implementation of simple classes.
6	Implementation of class hierarchies and polimorphism.
3	Use of enumerations and generics.
3	Use of Java library.
3	Programs using collections and I/O.
3	Final exam simulation.