UNIVERSITÀ DEGLI STUDI DI PALERMO

DEPARTMENT	Ingegneria
ACADEMIC YEAR	2019/2020
BACHELOR'S DEGREE (BSC)	INGEGNERIA CIBERNETICA
SUBJECT	PROGRAMMING
TYPE OF EDUCATIONAL ACTIVITY	A
AMBIT	50283-Matematica, informatica e statistica
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	PROGRAMMING - Corso: DIGITAL ENTERPRISE INNOVATION ENGINEERING PROGRAMMING - Corso: INGEGNERIA DELL'INNOVAZIONE PER LE IMPRESE DIGITALI
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	The final grade will range from 18/30 to 30/30 cum laude. The exam will consist of two examinations. With the first examination (conducted in the computer lab), the student will be asked to develop a Java program based on a textual description of the problem. Students achieving a score higher than 18/30 will be admitted to the oral discussion of the topics described in the course syllabus. The oral discussion will aim at verifying whether the student masters and understands the main concepts concerning object-oriented programming. The final mark, in 18/30 - 30/30 with honors, is calculated as average of the overall evaluation of the written and oral exams. According to the Dublin descriptors, the expected results will be assessed in relation to the final grade as follows: - from 18/30 to 20/30: mediocre or sufficient knowledge and understanding of the topics covered in the class, partial ability to apply the acquired knowledge to solve the proposed problems; partial autonomy of judgment, communication skills and the ability to learn. - from 21/30 to 23/30: sufficient or discrete knowledge and understanding of the topics covered, sufficient ability to apply the knowledge acquired for the resolution of the proposed problems, sufficient independence of judgment, communication skills and ability to learn. - from 24/30 to 26/30: discrete knowledge and understanding of the topics covered, discrete ability to apply the knowledge acquired for the resolution of the proposed problems, sufficient autonomy of judgment, communication skills and the ability to learn. - from 27/30 to 30/30 cum laude: good or excellent knowledge and understanding of the topics covered, good or excellent ability to apply the acquired knowledge for the resolution of the proposed problems, good or excellent judgment autonomy, enables communicative and ability to learn The minimum requirement for passing the exam is the demonstrated knowledge of notions related to classes and objects, inheritance, polymorphism, exception handling.
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 structures will also be studied with reference to their implementation in the Java library.

TEACHING METHODS	Lectures, computer laboratory sections
SUGGESTED BIBLIOGRAPHY	CAY S. HORSTMANN (2016). Core Java Volume I - Fundamentals (10th Edition), Prentice Hall 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.

SYLLABUS

Hrs	Frontal teaching
2	Introduction to Java programming language
4	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.
6	Inheritance and interfaces: inheritance, final modifier, Object class, inheritance and encapsulation, interfaces.
6	Polimorphism: method polimorphism, overload and override, data polimorphism.
4	Exceptions and assertions: exception handling in Java, exception propagation, introduction to assertions.
4	Collections: Collection interface, List, Queue and Deque, Map e SortedMap, algorithms
2	Generic types: creating generic types, type inference.
2	Enumeration and nested types: inner classes, anonymous classese, enumeration types.
4	Java library: String, Object, System, Runtime and Math classes, java.util package, Date-Time API.
3	Input/Output: character and byte stream, keyboard input, file management, objects serialization.

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