UNIVERSITÀ DEGLI STUDI DI PALERMO

DEPARTMENT	Ingegneria
ACADEMIC YEAR	2019/2020
MASTER'S DEGREE (MSC)	COMPUTER ENGINEERING
SUBJECT	FORMAL LANGUAGES AND COMPILERS
TYPE OF EDUCATIONAL ACTIVITY	B
AMBIT	50369-Ingegneria informatica
CODE	04761
SCIENTIFIC SECTOR(S)	ING-INF/05
HEAD PROFESSOR(S)	CHELLA ANTONIO Professore Ordinario Univ. di PALERMO
OTHER PROFESSOR(S)	
CREDITS	6
INDIVIDUAL STUDY (Hrs)	90
COURSE ACTIVITY (Hrs)	60
PROPAEDEUTICAL SUBJECTS	
MUTUALIZATION	
YEAR	1
TERM (SEMESTER)	2° semester
ATTENDANCE	Not mandatory
EVALUATION	Out of 30
TEACHER OFFICE HOURS	**CHELLA ANTONIO** Monday 09:00 11:00 DICGIM, edificio 6, III piano

DOCENTE: Prof. ANTONIO CHELLA

PREREQUISITES	- Algorithms and data structures; - Operating systems; - C and Java Programming.
LEARNING OUTCOMES	Learning outcomes according to the Dublin descriptors: - Objective 1: Knowledge and understanding The student will acquire the theoretical knowledge necessary to solve the problems related to the design and implementation of programming languages and the methodologies to the performance analysis. The student will thus study the typical case studies of programming languages and the main topics of current research. The course will include lectures; class discussions of case studies; seminars and panels. - Objective 2: Applying knowledge and understanding The student will acquire the working capabilities necessary to design and implement interpreters and compilers for programming languages. He/she will be able to create programming languages, to identify the problems, to formulate algorithms, to implement and evaluate the performances of the proposed solutions. The course will include sessions in the laboratory, or autonomously. - Objective 3: Making Judgments The student will acquire the necessary methodologies to implement and evaluate programming languages not previously discussed in the case studies by integrating all the notions obtained during the course. He/she will be able to analyze problem data at disposal, even if limited and incomplete, and to propose solutions tailored to the problem at hand. The student will be able to compare the strengths and weaknesses of the proposed solutions and to evaluate the performance of the proposed solutions. The course will include analysis and discussion of case studies; lectures and group sessions; class discussions and presentations performed by students team concerning their projects and implementations; preparation of a written essay. - Objective 4: Communication skills The student will be able to work in a team and to communicate with competence and correctness of language the issues related to the design, implementation, and evaluation of programming languages. The course will include team sessions in the laboratory on the design and implementation of programming languages; presentations and class discussions by the student's teams. - Objective 5: Learning skills The student will be able to autonomously learn and study specific difficult problems related to programming languages by the literature of the field. The course will include seminars, panels, and class discussion on the main research topics of programming languages.
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. - Assessment of Objective 1: Knowledge and understanding This objective will be assessed by an oral discussion concerning the theoretical topics of the syllabus. Objective 1 will count as 15% of the final grade. - Assessment of Objective 2: Applying knowledge and understanding This objective will be assessed by an oral discussion of programming languages case studies analyzed by the student during team sessions in the lab. Objective 2 will count as 15% of the final grade. - Assessment of Objective 3: Making Judgments This objective will be assessed by a discussion of an essay, written at home and in the lab, by the student together with his/her student team. The article will concern the design and implementation of a compiler for programming language performing assigned tasks. A live demo of the language will have to be shown by the student team. In particular, Objective 3 will be assessed by discussing, in particular, the design and implementation choices performed by the student team. Objective 3 will count as 30% of the final grade. - Assessment of Objective 4: Communication skills This objective will be assessed by the oral discussions concerning Objectives 1,2,3 and the analysis of the written essay concerning Objective 3. Objective 4 will count as 10% of the final grade. - Assessment of Objective 5: Learning skills This objective will be assessed using the discussion of the essay described in Objective 3. In particular, Objective 5 will be evaluated by discussing, in particular, the theories and techniques autonomously learned by the student team and employed in the implementation of the compiler for programming language. Objective 5 will count as 30% of the final grade.
EDUCATIONAL OBJECTIVES	Educational objectives are in agreement with the ACM/IEEE CS 2013 Body of Knowledge and they cover all or parts of the following Knowledge Units. PL/Object-Oriented Programming PL/Functional Programming PL/Basic Type Systems PL/Program Representation

	PL/Language Translation and Execution
TEACHING METHODS	The overall format of the course is: - Lectures - Lab sessions - Discussion classes
SUGGESTED BIBLIOGRAPHY	Ravi Sethi: Programming Languages, 2nd Edition, Pearson. Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman: Compilers, Principles, Techniques and Tools, Addison-Wesley.

SYLLABUS

Hrs	Frontal teaching
3	Overview: problems and progress of programming languages
3	Language description: syntactic structure
3	Structured programming
6	Types: data representation
3	Object-oriented programming
3	Elements of functional programming
3	Logic programming
3	Outlines of semantic methods
6	Introduction to compilers. Simple one-pass compilers.
3	Lexical analysis
3	Syntax analysis

Hrs	Practice
3	Characteristics of a programming language
3	Flex and Bison
3	The language ML