Università degli Studi di L'Aquila - Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica

Course catalogue

Page compiled on 18/11/2017

Programme of Course "Linguaggi di Programmazione e Compilatori"

- Code: F0151
- Type of course unit: Compulsory (Bachelor Degree in Computer Science curriculum General)
- Level of course unit: Undergraduate Degrees
- Semester: 1

Number of ects credits: (Bachelor Degree in Computer Science) 6 (workload 150 hours)

Teachers: Sergio Orefice (Sergio.Orefice@univaq.it)

1	**Course objectives**	The course is an introduction in compiler design and presents the main principles and techniques used during the compilation process, focusing on the analysis phases of a compiler
2	**Course content and learning outcomes (dublin descriptors)**	Topics of the module include: • Formal language basic notions. Introduction to compiling: the analysis-synthesis model, the phases of a compiler, interpreters • Lexical analysis: tokens-patterns-lexemes, input buffering, Thompson's algorithm • Syntax analysis: parsing top-down, parsing bottom-up. Nonrecursive predictive parsing • LR parsers: methods SLR, LALR, canonical LR • LR grammars and ambiguous grammars • Syntax-directed translation. Syntax-directed definitions: synthesized and inherited attributes, S-attributed definitions and L-attributed definitions • Translation schemes. Semantic analysis and type checking • Intermediate code generation. Three-address code: syntax and examples of generation On successful completion of this module, the student should : • acquire knowledge of the structure and the role of compilers; acquire knowledge and understanding of the principles and techniques used in the compilation process • be able to apply the methodologies learnt by the course to the development of small software projects regarding the construction of parts of compiler; • be able to assess the features of the lexical and syntactic components of a syntax-directed translation specification and choose the most suitable solution for its development; • understand and explain the relationships between tools and results from the formal language theory and their application to the analysis phases of compilers; • demonstrate capacity for reading and understanding other texts on related topics.
3	**Course prerequisites**	Knowledge of the basic concepts of the formal language theory and programming
4	**Teaching methodsand language**	Lectures and exercises **Language**: Italian **Reference textbooks** • A.V. Aho, R. Sethi, J.D. Ullmann, ***Compilers, principles, techniques and tools***. Addison-Wesley, Reading, Mass.
5	**Assessment methods**	Formative assessment: the formative assessment is performed via interaction between teacher and students during lectures. The students are encouraged to actively participate to the lectures by making questions and discussing the theoretical concepts presented and the solutions adopted in the developed examples. Summative assessment: written test on the subjects treated in the course. An optional mid-term written test will also be provided, which is meant to cover the first part of the course, in order to help the students to split the workload. The written test (lasting 2 hours) consists of a set of questions for the verification of theoretical/formal competences and for the verification of skills in understanding and solving significant exercises. Criteria of evaluation will be: the level of knowledge of the principles and techniques presented in the course, as well as the ability to apply them; the clarity and completeness of

		explanations.