



Programme of Course "Laboratorio di Architettura degli Elaboratori"

- Code: F11010
- Type of course unit: Compulsory (Bachelor Degree in Computer Science curriculum General)
- Level of course unit: Undergraduate Degrees
- Semester: 2

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

Teachers: Luca Forlizzi (Luca.Forlizzi@univaq.it)

1	<b>Course objectives</b>	This Course provides the students with the intermediate knowledge of imperative programming, of memory management techniques, of assembly language programming and related technologies, and of machine languages. At the end of the course the students will understand non-elementary programs written with an imperative programming language, will be able to realize constructs of a high level programming language with machine language instructions, and to understand and use the technology underlying a programming language. The students are expected to develop capacities for being attentive to details in algorithms implementation, to program portability, to the respect of programming languages standards and rules, as well as for looking for efficient implementations of algorithms.
2	<b>Course content and learning outcomes (dublin descriptors)</b>	<p>Topics of the module include:</p> <ul style="list-style-type: none"> <li>• Structured computed organization, computational models, programming languages and their standards</li> <li>• Imperative programming with high level languages oriented towards system programming</li> <li>• Basic and derived data types; data types representation</li> <li>• Memory management; low-level programming</li> <li>• Assembly language programming; translation of the main constructs of a high level programming language in an assembly language</li> <li>• Machine languages; assembly and disassembly processes</li> </ul> <p>On successful completion of this module, the student should :</p> <ul style="list-style-type: none"> <li>• have intermediate-level notions about imperative programming; have in-depth knowledge about memory management techniques, assembly language programming and related technologies, and about machine languages</li> <li>• be able to interpret non-elementary programs written with an imperative programming language; be able to implement constructs of a high level programming language with machine language instructions and interpret technical specifications of high-level programming languages, assembly languages and machine languages</li> <li>• be able to select, among different ways to implement programs, those leading to better performances</li> <li>• be able to discuss, with appropriate vocabulary, languages and computer architectures</li> <li>• be able to use an imperative language oriented towards system programming in subsequent courses about computer architectures; be able to see links between real computers and computational models as presented in theoretical computer science courses</li> </ul>
3	<b>Course prerequisites</b>	Knowledges: elementary knowledge of imperative programming, basics of computer systems architecture, elementary knowledge of mathematics, reading and understanding English language. Abilities: implementing correctly elementary algorithms, compiling and running simple programs on a computer
4	<b>Teaching methods and language</b>	<p>Lectures and exercises</p> <p><b>Language:</b> Italian</p> <p><b>Reference textbooks</b></p> <ul style="list-style-type: none"> <li>• G. Conte, A. Mazzeo, N. Mazzocca, P. Prinetto, <i>Architettura dei calcolatori</i>. CittàStudiEdizioni. 2015 (prima edizione).</li> </ul>

		<ul style="list-style-type: none"><li>• Kim N. King, <b>C Programming: a Modern Approach</b>. W.W.Norton &amp; Company. 2008 (second edition). <a href="http://knking.com/books/c2/index.html">http://knking.com/books/c2/index.html</a></li><li>• David A. Patterson, John L. Hennessy, <b>Struttura e progetto dei calcolatori</b>. Zanichelli. 2015 (quarta edizione, condotta sulla quinta edizione americana).</li></ul>
5	<b>Assessment methods</b>	<p>Pre-Assessment There is no formal pre-assessment, but Course pre-requisites are clearly stated on the website. Formative Assessment The formative assessment is performed via interaction between teacher and students during lectures. Students are involved in questioning and discussion, by means of open oral questions to the entire class. Summative Assessment The course grade is determined by a written test followed by an oral exam. It will be also provided an optional mid-term written test, which is meant to cover the first part of the course, in order to help the students to split the workload. The written tests consist in a series of exercises and open questions, to which the student has to provide a detailed extended answer, aimed to verify the practical abilities and level of knowledge acquired. The oral exam will occur within two weeks of the written test and will typically cover the areas of the written answers that need clarification, plus, possibly, additional subjects proposed by the teacher. Criteria of evaluation will be: level of knowledge and practical ability; the property of use of the technical/mathematical language; the clarity and completeness of explanations; capacity to make sensible decisions in the writing and translation of programs.</p>