

## SYLLABUS COURSE DESCRIPTION

<b>COURSE TITLE</b>	<b>Programming Paradigms</b>
<b>COURSE CODE</b>	76211
<b>SCIENTIFIC SECTOR</b>	INF/01
<b>DEGREE</b>	Bachelor in Computer Science
<b>SEMESTER</b>	1st
<b>YEAR</b>	2nd
<b>CREDITS</b>	6
<b>TOTAL LECTURING HOURS</b>	40
<b>TOTAL LAB HOURS</b>	20
<b>PREREQUISITES</b>	Students should have a solid mathematical foundation, good programming skills in an imperative or object -oriented language and be familiar with basic data structures and algorithms. These prerequisites are covered in the following courses: Analysis, Introduction to Programming, Programming Project, and Data Structures and Algorithms
<b>COURSE PAGE</b>	<a href="https://ole.unibz.it/">https://ole.unibz.it/</a>
<b>SPECIFIC EDUCATIONAL OBJECTIVES</b>	<p>Type of course: caratterizzanti          Scientific area: discipline informatiche</p> <p>Students will learn the key concepts and structures of the most popular programming paradigms, such as imperative, object-oriented, logic-oriented, functional and concurrent programming. They will practice to write small programs in different languages. Upon completion of the course, students shall have acquired basic programming skills in these languages and be able to judge strengths and weaknesses of different programming paradigms, in particular in the context of specific application domains.</p>
<b>LECTURER</b>	<a href="#">Elisa Marengo</a>
<b>SCIENTIFIC SECTOR OF THE LECTURER</b>	INF/01
<b>TEACHING LANGUAGE</b>	English

<b>OFFICE HOURS</b>	Wednesday 10:00-12:00 at another time after appointment by mail a few days in advance Faculty of Computer Science, Piazza Domenicani 3, office POS 2.02 <a href="mailto:elisa.marengo@unibz.it">elisa.marengo@unibz.it</a> +39 0471 016140
<b>TEACHING ASSISTANT</b>	Elisa Marengo Thomas Tschager
<b>OFFICE HOURS</b>	<a href="mailto:Thomas.tschager@unibz.it">Thomas.tschager@unibz.it</a> On Thursdays 13:00-14:00 or at another time after appointment by mail a few days in advance, Faculty of Computer Science, Piazza Domenicani 3, office POS 1.04
<b>LIST OF TOPICS COVERED</b>	<ul style="list-style-type: none"> <li>• Overview of programming paradigms</li> <li>• Basic elements of programming languages</li> <li>• OO programming</li> <li>• Logic programming</li> <li>• Concurrent programming</li> <li>• Functional programming</li> </ul>
<b>TEACHING FORMAT</b>	Frontal lecture
<b>LEARNING OUTCOMES</b>	<p><b>Knowledge and understanding</b></p> <ul style="list-style-type: none"> <li>• Know various programming paradigms and languages.</li> </ul> <p><b>Applying knowledge and understanding</b></p> <ul style="list-style-type: none"> <li>• Be able to develop small and medium size programs using different programming languages and paradigms.</li> </ul> <p><b>Ability to make judgments</b></p> <ul style="list-style-type: none"> <li>• Be able to evaluate strengths and weaknesses of different programming languages in specific application contexts.</li> </ul> <p><b>Ability to learn</b></p> <ul style="list-style-type: none"> <li>• Have developed learning capabilities to pursue further studies with a high degree of autonomy.</li> </ul>
<b>ASSESSMENT</b>	<p>The assessment of the course consists of a single written exam at the end that covers the whole course, plus the optional homework if handed in by the students.</p> <p>The written exam is structured in two parts:</p> <ul style="list-style-type: none"> <li>• 80% of the exam is to write small programs in each of the programming languages covered in the course;</li> <li>• 20% are questions about basic concepts.</li> </ul> <p>The first part verifies the ability to solve problems by developing small programs in different programming languages. The second part verifies the understanding of key concepts of different programming paradigms and languages.</p> <p>The optional homework consists in writing small programs that need to be handed in together with a small README file.</p>
<b>ASSESSMENT LANGUAGE</b>	English

<p><b>EVALUATION CRITERIA AND CRITERIA FOR AWARDING MARKS</b></p>	<p>There are no requirements for attending the final exam.</p> <p>The final course mark is computed from the mark of the written exam plus the mark of the homework if handed in by the student.</p> <p>The homework is only considered if it is marked higher than the written exam. In this case, the final course mark will be the average of the two marks; otherwise, the homework has no impact.</p> <p>The criteria for the evaluation of the exam and the homework are: correctness, completeness and clarity of the programs and answers.</p>
<p><b>REQUIRED READINGS</b></p>	<p>Lecture notes available on the course page</p>
<p><b>SUPPLEMENTARY READINGS</b></p>	<ul style="list-style-type: none"> <li>• Bruce A. Tate: Seven Languages in Seven Weeks Pragmatic Bookshelf, 2010 (recommended!)</li> <li>• Maurizio Gabrielli, Simone Martini: Programming Languages: Principles and Paradigms Springer, 2010 (optional)</li> <li>• Allen B. Tucker, Robert E. Noonan: Programming Languages: Principles and Paradigms (2nd ed.) McGraw Hill, 2007 (optional)</li> </ul>
<p><b>SOFTWARE USED</b></p>	<p>The following freely available software is used:</p> <ul style="list-style-type: none"> <li>• Ruby</li> <li>• Prolog</li> <li>• Erlang</li> <li>• Haskell</li> </ul>