


**Programma del Modulo "Laboratorio di Programmazione I"**

<ul style="list-style-type: none"> <li>• Codice: F1I056</li> <li>• Tipo di corso: Obbligatorio (Laurea in Informatica percorso Generale)</li> <li>• Livello del corso: Lauree di Primo Livello</li> <li>• Semestre: 1</li> </ul>		
Numero di crediti ECTS: (Laurea in Informatica) 6 (carico 150 ore)		
Docenti: Monica Nesi (monica.nesi@univaq.it)		
<b>1</b>	<b>Obiettivi del corso</b>	<p>Conoscenze: nozioni di base della programmazione imperativa ed orientata agli oggetti. Capacità: essere in grado di risolvere problemi semplici, saper implementare i relativi algoritmi correttamente in un linguaggio di programmazione strutturata, essere capace di strutturare dati complessi e le operazioni su tali dati, essere in grado di utilizzare il calcolatore per eseguire semplici programmi. Comportamenti attesi: interesse per l'attività di formalizzazione di problemi, dati e soluzioni in un linguaggio di programmazione, ed una costante attività di programmazione al calcolatore</p>
<b>2</b>	<b>Contenuti del corso e risultati formativi (descrittori di Dublino)</b>	<p>Gli argomenti trattati nel corso comprendono:</p> <ul style="list-style-type: none"> <li>• Algoritmi, programmi e linguaggi di programmazione. Diagrammi di flusso, struttura di un programma.</li> <li>• Tipi di dato base. Costanti, variabili, espressioni aritmetiche e booleane. Istruzione di assegnamento. Primitive di Input/Output. Strutture di controllo: sequenzializzazione, if-else, iterazione e ciclo.</li> <li>• Tipi di dato strutturati: array, stringhe.</li> <li>• Metodi. Struttura a blocchi e regole di scoping. Meccanismi di passaggio dei parametri. Metodi statici. Ricorsione e metodi ricorsivi.</li> <li>• Classi e oggetti: concetti base, creazione di oggetti e loro manipolazione. Metodi di istanza. Array di oggetti.</li> <li>• Ereditarietà e gerarchie. Polimorfismo e late binding. Eccezioni e loro gestione.</li> </ul> <p>Alla fine del corso, lo studente dovrebbe:</p> <ul style="list-style-type: none"> <li>• have profound knowledge of the basic concepts of imperative and object-oriented programming, have knowledge and understanding of algorithms for solving simple problems, have knowledge and understanding of the core of the programming language used for coding the algorithms, have knowledge of a simple environment to run programs;</li> <li>• understand and apply definitions as given by the syntax and the semantics of the programming language;</li> <li>• analyse and discuss different algorithms for solving a problem and different implementations of the same algorithm in the given language;</li> <li>• explain and illustrate the fundamental notions of basic and structured data types, control structures, methods definition and invocation, parameter passing, recursive methods, classes and objects, inheritance and exceptions;</li> <li>• demonstrate skill in problem-solving, demonstrate ability to use a (subset of a) programming language to code algorithms and test the solutions on computer, demonstrate capacity of abstraction and modularity.</li> </ul>
<b>3</b>	<b>Prerequisiti</b>	Nozioni matematiche di base (in particolare, insiemi e funzioni). Non è richiesta alcuna conoscenza di programmazione né la conoscenza di specifici linguaggi di programmazione.
<b>4</b>	<b>Modalità e lingua</b>	Lezioni ed esercizi

	<b>di insegnamento</b>	<b>Lingua:</b> Italiano <b>Testi/Bibliografia</b> <ul style="list-style-type: none"><li>• Cay Horstmann, <i>Concetti di informatica e fondamenti di JAVA</i>. Apogeo. 2007.</li><li>• Marco Bertacca e Andrea Guidi, <i>Programmare in Java</i>. McGraw-Hill. 2007.</li></ul>
<b>5</b>	<b>Metodi di accertamento</b>	La prova di esame consiste in una prova intermedia scritta + una prova conclusiva scritta + eventuale prova orale. Il voto finale della prove scritte, se maggiore o uguale a 18, puo' essere registrato senza sostenere la prova orale. La prova di esame negli appelli di recupero consiste di una prova scritta + eventuale prova orale.