

**SCHEDA DATI PER OFFERTA FORMATIVA PUBBLICA DI CUI AL PUNTO 1.2
DELLA CIRCOLARE MINISTERIALE N° 187 DELL'11 GIUGNO 2008**

Insegnamento: Docente titolare: Qualifica SSD di appartenenza Struttura di afferenza Telefono e-mail Orario di ricevimento Sito web docente	Linguaggi di Programmazione ad Oggetti 1 Giuliano Armano Professore di 2° fascia Sistemi di Elaborazione delle Informazioni (ING-INF/05) DIEE – Dip. di Ingegneria Elettrica ed Elettronica 070-675.5758 armano@diee.unica.it giovedì 11.30-13.30 http://iasc.diee.unica.it
Curriculum scientifico	<p>Le attività di ricerca di Giuliano Armano sono incentrate sui sistemi ad agenti software e sulle tecniche di soft-computing. Tali tecniche sono attivamente utilizzate in vari campi applicativi inerenti la bioinformatica, l'information retrieval e l'information filtering. Il Prof. Armano ha organizzato vari eventi internazionali (conferenze, workshop, special issue), in particolare nell'ambito dei sistemi ad agenti software e della bioinformatica.</p> <p>Pubblicazioni selezionate:</p> <p><i>G. Armano, M. Marchesi, and A. Murru, "A Hybrid Genetic-Neural Architecture for Stock Indexes Forecasting," Elsevier Int. Journal of Information Sciences, Shu-Heng Chen (ed.), Vol. 170(1), pp. 3-33, 2005.</i></p> <p><i>G. Armano, L. Milanese, and A. Orro, "Multiple Alignment Through Secondary Structure Information," IEEE Trans. on Nanobioscience, Vol. 4(3), pp. 207-211, Sept 2005.</i></p> <p><i>G. Armano, A. Orro, E. Vargiu. "MASSP3: A System for Predicting Protein Secondary Structure", Journal on Applied Signal Processing, EURASIP Journal on Applied Signal Processing (JASP), pp. 1-9, DOI 10.1155/ASP/2006/17195, 2006.</i></p> <p><i>G. Armano, F. Mascia, and E. Vargiu, "Using Taxonomic Domain Knowledge in Text Categorization Tasks", Int. Journal of Intelligent Control and Systems, Vol. 12(2), pp. 150-157 ISSN: 0218-2157. Special Issue on "Distributed Intelligent Systems", H. Zhu (ed.), June 2007.</i></p> <p><i>A. Addis, G. Armano, and E. Vargiu, "A Progressive Filtering Approach to Hierarchical Text Categorization", Communications of SIWN, Vol. 5, August 2008, pp. 28-32.</i></p>
Contenuto schematico del corso di insegnamento	<ol style="list-style-type: none"> 1. Introduzione alla tecnologia a oggetti (OOT) 2. Eredità e polimorfismo 3. OOT e linguaggio Java 4. OOT e linguaggio C++ 5. Approfondimenti sulla tecnologia a oggetti

<p>Obiettivi formativi e risultati attesi (secondo i descrittori di Dublino)</p>	<p>Alla fine del corso ci si attende che lo studente abbia sviluppato:</p> <p>1) [Indicatore di conoscenza e capacità di comprensione] la conoscenza dei fondamenti della programmazione orientata agli oggetti e la capacità di comprendere le diverse modalità con cui può essere declinata la tecnologia ad oggetti in funzione del linguaggio utilizzato.</p> <p>2) [Indicatore di conoscenza e capacità di comprensione applicate] la capacità di concettualizzare la soluzione di un problema in termini di oggetti interagenti e la capacità di realizzare semplici applicazioni software con un approccio che tenga conto delle caratteristiche del linguaggio di programmazione ad oggetti utilizzato.</p> <p>3) [Indicatore autonomia di giudizio] la capacità di definire soluzioni che tengano conto sia delle caratteristiche specifiche del problema sia dell'espressività e delle particolarità del linguaggio di programmazione ad oggetti utilizzato.</p> <p>4) [Indicatore abilità comunicative] la capacità di identificare e descrivere, utilizzando una corretta terminologia, un sistema in termini di oggetti interagenti.</p> <p>5) [Indicatore capacità di apprendere autonomamente] l'acquisizione di schemi e strumenti utili per la concettualizzazione di problemi software, nonché la capacità di estendere autonomamente le proprie conoscenze sui linguaggi di programmazione.</p>
<p>Articolazione del corso</p>	<p>1. Introduzione alla tecnologia a oggetti (OOT) [2T]</p> <p>Riepilogo di nozioni propedeutiche: specifica sintattica, semantica e pragmatica dei dati, tipi di dati astratti (ADT)</p> <p>Concetti caratterizzanti l'OOT: ADT, eredità e polimorfismo</p> <p>Terminologia di riferimento: tipi/classi, operazioni/metodi, oggetti, dispatching, binding</p> <p>2. Eredità e polimorfismo [6T]</p> <p>Forme di eredità (singola e multipla), shadowing, overriding, redefinition.</p> <p>Modalità di derivazione classe-sottoclasse: specializzazione, specificazione, estensione, combinazione, costruzione, varianza, limitazione, generalizzazione</p> <p>Polimorfismo secondo Cardelli e Wegner: universale (parametrico e per inclusione) e ad-hoc (overloading e coercion)</p> <p>3. OOT e linguaggio Java (con esercitazioni) [10T+8E]</p> <p>Introduzione e generalità: Java Virtual Machine (JVM), thread,</p>

	<p>garbage collector, struttura di un programma Java</p> <p>Eredità e polimorfismo in Java: definizione di classe, l'oggetto "this", tipi primitivi e classi Wrapper, classi e interfacce, package, modificatori di accesso, costruttori, catena di eredità, l'oggetto "super", binding e dispatching, membri "static".</p> <p>Altre caratteristiche distintive del linguaggio Java: gestione delle eccezioni, operazioni di input/output, programmazione concorrente e thread, modello di programmazione basato su eventi e AWT, networking</p> <p>4. OOT e linguaggio C++ (con esercitazioni) [10T+8E]</p> <p>Introduzione e generalità: principali differenze rispetto al linguaggio C, struttura di un programma in C++.</p> <p>Eredità e polimorfismo in C++: definizione di classe, il puntatore "this", classi, namespace, modificatori di accesso, costruttori e distruttori, catena di eredità, binding e dispatching, metodi virtuali, membri "static".</p> <p>Raccomandazioni per le buone pratiche di programmazione in C++: costruttori, costruttori di copia, distruttori, assegnamento, verifica di uguaglianza e inizializzazione di oggetti.</p> <p>Altre caratteristiche distintive del linguaggio C++: overloading di operatori, operatori di cast definiti dall'utente, argomenti di default, dichiarazioni "friend", gestione delle eccezioni, operazioni di input/output, template (di funzione e di classe), iteratori</p> <p>Standard Template Library (STL): introduzione, principali classi contenitore: vector, string, list, deque, map.</p> <p>5. Approfondimenti sulla tecnologia a oggetti [6T]</p> <p>Definizione informale dei concetti di subclassing e subtyping Principio di sostituibilità di Liskov (LSP) e subtyping Invarianti di classe e pre-/post-condizioni di operazioni Varianza / controvarianza dei parametri Caratterizzazione delle principali modalità di derivazione classe-sottoclasse in termini di LSP Confronto tra alcuni linguaggi Object Oriented: C++, Java, Python, Smalltalk (cenni)</p>
Propedeuticità	Fondamenti di Informatica 2
Anno di corso e semestre	2° anno, 2° sem.
Testi di riferimento	<p>- <u>parte OOT</u></p> <p>T. Budd, An Introduction to Object-Oriented Programming,</p>

	<p>Pearson Addison-Wesley.</p> <p>- <u>parte sul linguaggio Java</u> [in alternativa]</p> <p>M. Bertacca, A. Guidi – Introduzione a Java – McGraw-Hill.</p> <p>K. Arnold, J. Gosling – Java, Didattica e Programmazione, Addison-Wesley.</p> <p>- <u>parte sul linguaggio C++</u> [in alternativa]</p> <p>B. Stroustrup, Il linguaggio C++, Addison Wesley Masson, Milano.</p> <p>S.B. Lippman, and S. Lajoie, C++ Corso di Programmazione, Addison Wesley Masson, Milano.</p>
Modalità di erogazione dell'insegnamento	Tradizionale
Modalità di frequenza	Obbligatoria
Metodi di valutazione	Prova scritta/prove in itinere
Organizzazione della didattica	50 ore, di cui 34 ore di lezione e 16 ore di esercitazione