

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

<b>Insegnamento:</b> <b>Docente titolare:</b> <b>Qualifica</b> <b>SSD di appartenenza</b> <b>Struttura di afferenza</b> <b>Telefono</b> <b>e-mail</b> <b>Orario di ricevimento</b> <b>Sito web docente</b>	Linguaggi di Programmazione ad Oggetti 2 Ing. Sandro Pinna, PhD Assegnista di ricerca post-doc ING-INF/05 Dipartimento di ingegneria Elettrica ed Elettronica 070 675-5761 <a href="mailto:sandro.pinna@diee.unica.it">sandro.pinna@diee.unica.it</a> su appuntamento <a href="http://agile.diee.unica.it">http://agile.diee.unica.it</a>
<b>Curriculum scientifico</b>	<p>Sandro Pinna è assegnista di ricerca post-doc e professore a contratto del corso di Linguaggi di programmazione ad oggetti 2 presso il Dipartimento di Ingegneria Elettrica ed Elettronica dell'Università di Cagliari.</p> <p>Attualmente si occupa dello studio delle metriche di qualità dei sistemi software e della realizzazione di appositi tool per l'estrazione di dati sulla difettosità e sulla qualità di progetti open source. La qualità dei sistemi viene misurata sia in termini di metriche tradizionali (per esempio le metriche CK) sia mediante l'applicazione ai sistemi software della teoria delle reti complesse. Un altro filone di ricerca riguarda le metodologie agili per la produzione del software</p> <p>Principali pubblicazioni:</p> <p>CONCAS G, MARCHESI M., PINNA S, SERRA N (2007). Power-Laws in a Large Object-Oriented Software System. IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, vol. 33; p. 687-708, ISSN: 0098-5589, doi: 10.1109/TSE.2007.1019</p> <p>CONCAS G, LOCCI M.F, MARCHESI M., PINNA S, TURNU I (2006). Fractal Dimension in Software Networks. EUROPHYSICS LETTERS, vol. 76; p. 1221-1227, ISSN: 0295-5075, doi: 10.1209/epl/i2006-10384-1</p> <p>CONCAS G, MARCHESI M., PINNA S, SERRA N (2006). On the suitability of Yule process to stochastically model some properties of object-oriented systems. PHYSICA. A, vol. 370; p. 817-831, ISSN: 0378-4371, doi: 10.1016/j.physa.2006.02.024</p> <p>M. Angioni, D. Carboni, S. Pinna, R. Sanna, N. Serra, A. Soro, Integrating XP project management in development environments, Journal of Systems Architecture 52 (2006) 619-626</p>
<b>Contenuto schematico del corso di insegnamento</b>	Architettura a livelli di una applicazione enterprise. Pattern per lo sviluppo di applicazioni enterprise. La piattaforma JEE. Mapping di una struttura ad oggetti su un DB relazionale: Java Persistence API.
<b>Obiettivi formativi e risultati attesi (secondo i descrittori di Dublino)</b>	· Conoscenza e capacità di comprensione: conoscere i pattern per lo sviluppo di applicazioni enterprise e capacità di comprendere le implicazioni del loro utilizzo sulla qualità e sulla

	<p>manutenibilità dei sistemi.</p> <ul style="list-style-type: none"> <li>· Conoscenza e capacità di comprensione applicate: capacità di individuare le soluzioni software attraverso le quali implementare i pattern</li> <li>· Autonomia di giudizio: sviluppare la capacità di scegliere la migliore soluzione software per l'implementazione di un pattern in ogni contesto applicativo</li> <li>· Abilità comunicative: capacità di esprimere in maniera semplice concetti tecnici. Capacità di sviluppare software in ambienti collaborativi.</li> <li>· Capacità di apprendere: Capacità di apprendere da diverse fonti e flessibilità nell'utilizzo di framework open source in differenti linguaggi di programmazione</li> </ul>
<b>Articolazione del corso</b>	<p>Applicazioni Enterprise. I livelli di una applicazione enterprise. I pattern per lo sviluppo di applicazioni enterprise. La piattaforma JEE. Le servlet: ripasso sul protocollo http, i metodi doGet e doPost, generare una pagina di risposta tramite il metodo println, forward e redirect, i filtri, deployment su un application server. Gestire una sessione. Chiamare una servlet da un'altra servlet. (6 ore di teoria e 4 di esercitazione)</p> <p>Le Java Server Pages (JSP): Scriptlet, expression, declaration, Oggetti impliciti. Chiamare una JSP da una servlet. Le Java Standard Tag Libraries (JSTL). (4 ore di teoria e 4 di esercitazione)</p> <p>Mappare una struttura ad oggetti su un DB relazionale. Java Database Connectivity. Il pattern DAO (Data Access Object). Strumenti ORM. Java Persistence API (JPA). Cenni sulle transazioni. (5 ore di teoria e 5 di esercitazione)</p> <p>Pattern MVC. Pattern Front Controller. Il framework Struts2: action, tag libraries, validazioni lato client e lato server. (8 ore di teoria e 6 ore esercitazione)</p> <p>Il pattern Dependency Injection. Il framework Spring (cenni). (4 ore di teoria)</p> <p>Le principali vulnerabilità delle applicazioni web. Open Web Application Security Project (OWASP). (2 ore di teoria e 2 ore esercitazione)</p>
<b>Propedeuticità</b>	Buona conoscenza di almeno un linguaggio di programmazione ad oggetti e dei principi dell'ingegneria del software.
<b>Anno di corso e semestre</b>	2° anno/ 2° sem.
<b>Testi di riferimento</b>	<p>Martin Fowler-Patterns of Enterprise Application Architecture-Addison Wesley</p> <p>Dispense nel sito del corso</p>
<b>Modalità di erogazione dell'insegnamento</b>	Tradizionale
<b>Modalità di frequenza</b>	Facoltativa ma consigliata
<b>Metodi di valutazione</b>	L'esame consiste nello sviluppo di un progetto e nella discussione finale dello stesso.
<b>Organizzazione della didattica</b>	50 ore, di cui 29 ore di lezione e 21 ore di esercitazione.