

<b>Principali informazioni sull'insegnamento</b>	<b>A.A. 2018-2019</b>
Titolo insegnamento	Ingegneria del Software
Corso di studio	Informatica
Crediti formativi	9
Denominazione inglese	Software Engineering
Obbligo di frequenza	No
Lingua di erogazione	Italiano

<b>Docente responsabile</b>	Nome Cognome	Indirizzo Mail
	Filippo Lanubile	filippo.lanubile@uniba.it
Luogo ed Orario di Ricevimento	Dip. Informatica 6° Piano	Venerdì dalle 12:00 alle 14:00

<b>Dettaglio credi formativi</b>	Ambito disciplinare	SSD	Crediti
	Informatico	INF/01	9

<b>Modalità di erogazione</b>	
Periodo di erogazione	Secondo Semestre
Anno di corso	Secondo Anno
Modalità di erogazione	Lezioni frontali Esercitazioni in aula e laboratorio

<b>Organizzazione della didattica</b>	
Ore totali	225
Ore di corso	71 (56 lezioni frontali e 15 esercitazioni/laboratorio)
Ore di studio individuale	154 (119 lezioni frontali, 10 esercitazioni/laboratorio, 25 progetto)

<b>Calendario</b>	
Inizio attività didattiche	25 febbraio 2018
Fine attività didattiche	31 maggio 2018

<b>Syllabus</b>	
Prerequisiti	<p>Deve essere stato colmato l'eventuale debito formativo secondo quanto previsto dal Regolamento Didattico del Corso di Studi.</p> <p>È fortemente consigliato il superamento degli esami dei seguenti insegnamenti del I anno: Programmazione., Laboratorio di Informatica, Linguaggi di Programmazione.</p>
Risultati di apprendimento previsti (declinare rispetto ai Descrittori di Dublino) (si raccomanda che siano coerenti con i risultati di apprendimento del CdS, compreso i risultati di	<ul style="list-style-type: none"> <li><i>Conoscenza e capacità di comprensione</i> Conoscere principi fondanti dell'ingegneria del software Conoscere le diverse tipologie di applicazioni software Conoscere le differenze tra i processi di sviluppo del software, in particolare tra modello waterfall e modello</li> </ul>

<p>apprendimento trasversali)</p>	<p>agile</p> <p>Conoscere i diversi aspetti modellabili del software e le possibili finalità della modellazione</p> <p>Conoscere le soluzioni di successo a problemi noti e ricorrenti nella progettazione del software</p> <ul style="list-style-type: none"> <li>• <i>Conoscenza e capacità di comprensione applicate</i> Sviluppare competenze applicabili allo sviluppo ed evoluzione di sistemi software, aventi caratteristiche di qualità, economicità e tempestività. Sviluppare la capacità di creare modelli, sia con prospettiva concettuale che con prospettiva software. Acquisire familiarità con lo sviluppo di sistemi software la cui realizzazione richiede lo sforzo congiunto e prolungato di un team di lavoro.</li> <li>• <i>Autonomia di giudizio</i> Mostrare di aver acquisito autonomia di giudizio sulle scelte relative allo sviluppo dei sistemi software.</li> <li>• <i>Abilità comunicative</i> Mostrare di essere in grado di comunicare in modo appropriato le caratteristiche di prodotto e di processo nello sviluppo del software.</li> <li>• <i>Capacità di apprendere</i> Mostrare di aver sviluppato capacità di intraprendere in autonomia ulteriori approfondimenti su argomenti attinenti all'ingegneria del software.</li> </ul>
<p>Contenuti di insegnamento</p>	<ul style="list-style-type: none"> <li>• Introduzione all'ingegneria del software</li> <li>• Processi software e sviluppo agile; Scrum</li> <li>• Controllo di versione; git; GitHub; GitHub Flow</li> <li>• Modellazione con UML</li> <li>• Analisi dei requisiti; modellare il dominio e il workflow</li> <li>• Automazione dello sviluppo e del rilascio</li> <li>• System design; stili architetturali</li> <li>• OO design; modellare la struttura e il comportamento; principi e design pattern</li> <li>• Testing; analisi statica del codice; automazione del test di unità: JUnit</li> <li>• Manutenzione ed evoluzione</li> </ul>

<p><b>Programma</b></p>	
<p>Testi di riferimento</p>	<p>- Bernd Bruegge, Dutoit Allen. <i>Object-Oriented Software Engineering Using UML, Patterns, and Java</i>. Pearson New Int. Ed. (Inglese)</p> <p>Altri libri consigliati:</p> <ul style="list-style-type: none"> <li>- Martin Fowler. <i>UML Distilled</i>. Pearson.</li> <li>- Craig Larman. <i>Applicare UML e i pattern</i>. Quarta edizione. Pearson.</li> </ul>

Note ai testi di riferimento	I libri di testo sono integrati con gli appunti presi a lezione, con le slide del docente e puntatori a risorse online disponibili sulla piattaforma ADA.
Metodi didattici	Lezioni frontali supportate da slide, esercitazioni guidate dal docente in aula, laboratorio di tipo Bring-Your-Own-Device BYOD con assegnazioni di sviluppo e modellazione verificate in aula.
Metodi di valutazione (indicare almeno la tipologia scritto, orale, altro)	<p>Prova di laboratorio e prova scritta.</p> <ul style="list-style-type: none"> <li>- La prova di laboratorio consiste in esercizi di modellazione UML, in modalità forward o reverse engineering a partire da codice Java, utilizzando git e GitHub.</li> <li>- La prova scritta consiste nel rispondere a un questionario contenente domande a risposta chiusa o aperta.</li> <li>- Sia la prova di laboratorio che la prova scritta sono valutate in trentesimi. Il voto finale è la media aritmetica delle due prove.</li> <li>- La prova di laboratorio e la prova scritta devono essere sostenute e superate nello stesso appello.</li> </ul> <p>Per gli studenti che frequentano attivamente è possibile svolgere un'attività di progetto come esonero dalle prove di laboratorio e scritta.</p> <p>Il progetto consiste nel completare in gruppo, durante lo svolgimento delle lezioni, lo sviluppo di un sistema software assegnato dal docente, seguendo un processo anche questo dettato dal docente.</p> <p>Un progetto valutato positivamente non può essere sottoposto a ulteriore valutazione. La valutazione positiva di un progetto è valida per tutti gli appelli dell'anno accademico corrente. La valutazione dell'attività di progetto è in trentesimi ed è condivisa dall'intero gruppo di lavoro. La valutazione individuale è quindi corretta con punti di bonus o malus in relazione al contributo dato al gruppo e all'intera classe.</p>
Criteri di valutazione (per ogni risultato di apprendimento atteso su indicato, descrivere cosa ci si aspetta lo studente conosca o sia in grado di fare e a quale livello al fine di dimostrare che un risultato di apprendimento è stato raggiunto e a quale livello)	Saranno valutati i risultati di apprendimento previsti così come descritti precedentemente.
Altro	