

Principali informazioni sull'insegnamento	A.A. 2020-2021
<i>Titolo insegnamento</i>	Ingegneria del Software – corso A (lettere A-L)
<i>Corso di studio</i>	Informatica e Tecnologie per la Produzione del Software
<i>Crediti formativi</i>	12 CFU
<i>Denominazione inglese</i>	Software Engineering
<i>Obbligo di frequenza</i>	No
<i>Lingua di erogazione</i>	Italiano

Docente responsabile	<i>Nome Cognome</i>	<i>Indirizzo Mail</i>
	Danilo Caivano	danilo.caivano@uniba.it
<i>Luogo ed Orario di Ricevimento</i>	Dip. Informatica VI Piano	lunedì dalle 10:30 alle 11:30

Dettaglio crediti formativi	<i>Ambito disciplinare</i>	<i>SSD</i>	<i>Crediti</i>
Lezioni frontali (9 CFU tipo T1)	Informatico	ING-INF/05 - Sistemi di Elaborazione delle Informazioni	9
Progetto (3 CFU di tipo T3)	Informatico	ING-INF/05 - Sistemi di Elaborazione delle Informazioni	3

Modalità di erogazione	
<i>Periodo di erogazione</i>	Secondo Semestre
<i>Anno di corso</i>	Secondo Anno
<i>Modalità di erogazione</i>	Lezioni frontali (9 CFU) Progetto (3 CFU)

Organizzazione della didattica	
<i>Ore totali</i>	300
<i>Ore di corso</i>	72 ore di lezioni frontali
<i>Ore di studio individuale</i>	228 (153 ore di studio individuale, 75 ore di progetto)

Calendario	
<i>Inizio attività didattiche</i>	1 marzo 2021
<i>Fine attività didattiche</i>	4 giugno 2021

Syllabus	
<i>Prerequisiti</i>	Lo studente deve avere familiarità con almeno un linguaggio di programmazione e con le strutture di dati fondamentali. Non ci sono specifiche propedeuticità da rispettare oltre quelle indicate nel Regolamento Didattico e Manifesto degli Studio.

<p><i>Risultati di apprendimento previsti (declinare rispetto ai Descrittori di Dublino) (si raccomanda che siano coerenti con i risultati di apprendimento del CdS, riportati nei quadri A4a, A4b e A4c della SUA, compreso i risultati di apprendimento trasversali)</i></p>	<ul style="list-style-type: none"> • <i>Conoscenza e capacità di comprensione</i> Il principale risultato di apprendimento previsto è la conoscenza relativa a principi, paradigmi, metodologie, tecniche e tecnologie fondamentali per l'analisi e progettazione in team di sistemi software di medie-grandi dimensioni supportati da strumenti allo stato della pratica. Tali conoscenze mirano anche a fornire allo studente le competenze necessarie nella produzione e manutenzione di software applicativo per le applicazioni d'impresa. Lo studente acquisisce tale conoscenza sia attraverso le lezioni frontali e la partecipazione a seminari tematici erogati durante il corso, sia attraverso esercitazioni che gli consente di mettere in pratica e verificare quanto appreso, acquisendo così consapevolezza della capacità di comprensione e di come migliorare l'applicazione delle tecniche apprese. • <i>Conoscenza e capacità di comprensione applicate</i> Per consentire allo studente di applicare le conoscenze per lo sviluppo (produzione e manutenzione) delle Applicazioni d'Impresa, si svolgono in aula sia esercitazioni individuali che collettive. Allo studente è richiesto di sviluppare un progetto, nel quale è necessario applicare i principi di ingegneria del software, le metodologie e le tecniche presentate a lezione, selezionando quelle più adeguate per lo specifico caso. La valutazione di tale progetto contribuisce alla valutazione finale dello studente e quindi al voto conseguito all'esame di profitto. • <i>Autonomia di giudizio</i> Obiettivo importante del corso è che lo studente raggiunga una significativa autonomia nell'operare le opportune scelte durante l'analisi, la progettazione e lo sviluppo del sistema software oggetto del progetto. Le esercitazioni che si svolgono durante il corso contribuiscono al raggiungimento di tale autonomia grazie anche alla discussione di tali scelte con il docente. L'autonomia di giudizio è parte della valutazione finale dello studente e tiene conto delle discussioni avvenute durante le lezioni, delle esercitazioni e della presentazione del progetto. • <i>Abilità comunicative</i> Lo studente è stimolato a lavorare in gruppo ed è invitato a illustrare il risultato di esercizi svolti, autonomamente o in gruppo, con l'obiettivo di sviluppare le sue abilità comunicative. A tale scopo, allo studente è richiesto di sviluppare un progetto in gruppo in cui applicare i principi, le metodologie e le tecniche che ha appreso, selezionando quelle ritenute più appropriate (in base alla propria autonomia di giudizio). La presentazione e discussione di tale progetto è parte della prova orale d'esame e consente allo studente di mostrare le proprie abilità comunicative. • <i>Capacità di apprendere</i> Per stimolare la capacità di apprendere in modo autonomo, allo studente è richiesto di approfondire specifici argomenti non trattati in dettaglio dal docente oppure è invitato a partecipare a seminari tenuti da altri docenti, interni o in visita al dipartimento, sui quali lo studente deve poi presentare durante le lezioni, e riportare in sede d'esame.
--	---

<p><i>Contenuti di insegnamento</i></p>	<p>L'insegnamento di Ingegneria del Software riguarda l'analisi, il progetto e la realizzazione di sistemi software applicando i principi dell'Ingegneria del Software, nonché metodologie e tecniche di sviluppo di sistemi software. Ciò include la costruzione di una applicazione d'impresa, dalla raccolta dei requisiti alla realizzazione della stessa. In particolare saranno presentati i contenuti descritti di seguito.</p> <p>Introduzione</p> <ul style="list-style-type: none"> - Visione d'insieme - I tipi di prodotti software - Processi di sviluppo software - Qualità dei prodotti - Problemi dell'ingegneria del software <p>Principi dell'Ingegneria del Software</p> <ul style="list-style-type: none"> - Applicabilità dei principi - Rigore e formalità, Separazione degli interessi, Modularità, Astrazione, Generalità, Incrementalità <p>Analisi</p> <ul style="list-style-type: none"> - Concetti generali - Specifiche dei Requisiti - Specifiche Software <p>Processi Agili</p> <ul style="list-style-type: none"> - Sviluppo Agile del Software - Metodologia SCRUM <p>Progetto Software</p> <ul style="list-style-type: none"> - Concetti Generali - Elementi di Base sui Processi - Linee guida di progetto (Information Hiding) - Processo di progettazione SW <p>Linguaggio di modellazione di un sistema software – UML</p> <ul style="list-style-type: none"> - Overview - Diagramma dei casi d'uso: casi d'uso, scenari, relazioni - Diagramma delle classi: classi, oggetti, relazioni - Diagramma di sequenza - Diagramma di comunicazione - Diagramma delle attività - Diagramma di package - Diagramma delle componenti - Diagramma di deployment - UML per il web: Web Application Extension - Stereotipi UML: Approccio BCE (Boundary – Control -Entity) - Esempi di modellazione UML <p>Software Project Management</p> <ul style="list-style-type: none"> - Project Management - Progetti software e complessità - Cenni sugli strumenti di Project Management
---	---

	<p>Stili Architeturali</p> <ul style="list-style-type: none"> - Principi Generali - Stili Architeturali - Object Oriented <p>Design Pattern</p> <ul style="list-style-type: none"> - Pattern di creazione - Pattern strutturali - Pattern comportamentali - Pattern J2EE <p>Privacy Oriented Software Development (POSD)</p> <ul style="list-style-type: none"> - Privacy by Design: Principi Guida - Fondamenti metodologici - utilizzo di Design Pattern - Privacy Knowledge Base (PKB) <p>Strumenti di Supporto allo sviluppo</p> <ul style="list-style-type: none"> - Application Lifecycle Management (ALM) - Configuration Management - Quality Management
--	--

Programma	
<i>Testi di riferimento</i>	<ol style="list-style-type: none"> 1. Carlo Ghezzi, Medhi Jazayeri, Dino Mandrioli "Ingegneria del Software - Fondamenti e Principi, 2a edizione" Pearson Prentice Hall, 2004. 2. Martin Fowler "UML distilled. Guida rapida al linguaggio di modellazione standard" (4 ed.). Pearson Addison Wesley, 2010.
<i>Note ai testi di riferimento</i>	<p>I testi di riferimento sono integrati con slide, dispense del docente e altro materiale didattico messi a disposizione degli studenti sulla piattaforma di e-learning usata dal CdS.</p> <p>Testi consigliati per approfondire specifici argomenti:</p> <ul style="list-style-type: none"> - Roger S. Pressman "Principi di ingegneria del software". McGraw-Hill, 2008. - Jim Arlow, Ila Neustadt "UML 2 e Unified Process – Analisi e progettazione Object-Oriented, 2a edizione", McGraw-Hill, 2014 - Steven John Metsker, "Design pattern in Java : manuale pratico". Pearson : Addison Wesley, 2003.
<i>Metodi didattici</i>	<p>Lezioni frontali con l'ausilio di slide che riportano esempi per illustrare gli argomenti trattati.</p> <p>Esercitazioni pratiche sull'utilizzo dei vari principi e tecniche presentate a lezione attraverso esercizi da svolgere singolarmente.</p> <p>Un progetto da svolgere in gruppo utilizzando Redmine quale strumento di Application Lifecycle Management.</p> <p>Utilizzo della piattaforma di e-learning del Dipartimento di Informatica per la distribuzione del materiale e per le interazioni tra docenti e studenti durante e dopo il corso.</p>

<p><i>Metodi di valutazione (indicare almeno la tipologia scritto, orale, altro)</i></p>	<p>La verifica dei risultati formativi raggiunti avviene durante l'esame finale, che prevede:</p> <ul style="list-style-type: none"> • un colloquio orale in cui si presenta e si discute il progetto sviluppato in gruppo e si verificano le competenze acquisite durante il corso e le capacità espositive dello studente. • una prova scritta con domande a risposta multipla, chiuse e/o aperte. <p>Per gli studenti frequentanti sono previste le seguenti facilitazioni:</p> <ul style="list-style-type: none"> - bonus punteggio a valere sulla valutazione del progetto per gli studenti che svolgono positivamente le esercitazioni sul progetto/caso di studio; - prove intermedie in itinere esoneranti la prova scritta. <p>Le facilitazioni per lo studente frequentante sono valide solo per i primi tre appelli (sessione estiva) dell'anno accademico corrente.</p>
<p><i>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)</i></p>	<p>Per accertare la conoscenza acquisita dallo studente e le sue autonomie di giudizio, abilità comunicativa e capacità di apprendere:</p> <ul style="list-style-type: none"> • si valuta il progetto svolto in gruppo tenendo conto di come sono stati applicati i principi e le metodologie, dell'appropriatezza delle tecniche usate, dell'originalità delle soluzioni, della chiarezza e della capacità di sintesi che risultano dalla documentazione prodotta; • Si valuta la presentazione del progetto per verificare le competenze acquisite dallo studente e la sua capacità di sintesi nonché la chiarezza di esposizione, la capacità di fare confronti significativi tra metodologie, tecniche e tecnologie diverse adottate e riportare un proprio giudizio critico; • Si valuta la prova scritta con domande a risposta multipla, chiuse e/o aperte, per accertare le conoscenze di base dello studente. <p>Il voto del progetto e la sua presentazione concorrono a circa il 60% del voto complessivo dell'esame e la prova scritta al rimanente 40%.</p>
<p><i>Altro</i></p>	<p>Il progetto è assegnato all'inizio del semestre di lezioni. La consegna del progetto deve avvenire 3 giorni lavorativi prima della data dell'appello.</p>