UNIVERSITÀ DEGLI STUDI DI PALERMO

DIPARTIMENTO	Matematica e Informatica
ANNO ACCADEMICO OFFERTA	2023/2024
ANNO ACCADEMICO EROGAZIONE	2025/2026
CORSO DILAUREA	INFORMATICA
INSEGNAMENTO	INGEGNERIA E SICUREZZA DEL SOFTWARE
TIPO DI ATTIVITA'	B
AMBITO	50166-Discipline Informatiche
CODICE INSEGNAMENTO	20632
SETTORI SCIENTIFICO-DISCIPLINARI	INF/01
DOCENTE RESPONSABILE	ROMBO SIMONA ESTER Professore Ordinario Univ. di PALERMO
ALTRI DOCENTI	
CFU	6
NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE	78
NUMERO DI ORE RISERVATE ALLA DIDATTICA ASSISTITA	72
PROPEDEUTICITA'	
MUTUAZIONI	
ANNO DI CORSO	3
PERIODO DELLE LEZIONI	1° semestre
MODALITA' DI FREQUENZA	Facoltativa
TIPO DI VALUTAZIONE	Voto in trentesimi
ORARIO DI RICEVIMENTO DEGLI STUDENTI	**ROMBO SIMONA ESTER** Lunedì 9:30 13:30 In presenza: Via Archirafi 34, Piano II, Stanza 220 - Telematico: via Microsoft Teams o altro canale - In entrambi i casi, e' consigliabile prenotarsi tramite email alla docente

DOCENTE: Prof.ssa SIMONA ESTER ROMBO

PREREQUISITI	Programmazione e programmazione orientata agli oggetti. Progettazione e interrogazione di database.
RISULTATI DI APPRENDIMENTO ATTESI	Lo studente acquisira: Conoscenze di base e capacità di comprensione sull'ingegneria del software e conoscenza della metodologia Agile e dell'uso di UML in tale ambito. Capacita' di applicare metodologie Agili alla progettazione di un applicativo con grado di complessità simile a quello dei contesti industriali, con eventuali applicazioni nell'ambito dell'Intelligenza Artificiale; conoscenza di design pattern di progettazione e architetturali; capacita' di individuare l'utilità e l'applicabilità dei design pattern al progetto di un software. Autonomia di giudizio nel formulare le scelte progettuali. Abilità comunicative nella presentazione delle scelte compiute e della soluzione proposta, dimostrate anche attraverso la loro esposizione di fronte un pubblico di utenti con pari o superiore esperienza. Capacità di apprendere nuove metodologie per la progettazione e l'implementazione del software.
VALUTAZIONE DELL'APPRENDIMENTO	L'esame è composto da due parti: (1) stesura di un progetto che rispecchi il livello di difficoltà delle applicazioni del mondo reale, e (2) prova orale. La stesura del progetto viene svolta in team, in parte durante alcune ore di laboratorio e in parte durante il proprio tempo di studio individuale. La valutazione del progetto avviene attraverso delle revisioni ed una presentazione finale, a cui tutto il team deve partecipare, e che hanno lo scopo di verificare le capacita' acquisite dallo studente nell'applicazione dei principi dell'Ingegneria del Software, lavoro in team, autonomia di giudizio. La valutazione del progetto è espressa in trentesimi. La prova orale verte sui temi trattati durante il corso e la valutazione, anche in questo caso, viene espressa in trentesimi. La votazione finale è data dalla media fra i due voti ottenuti nelle due parti che compongono la prova d'esame. In particolare, una valutazione superiore a 28/30 indica un'ottima conoscenza degli argomenti del corso e un'eccellente capacità di progettare seguendo i principi appresi. Nel caso del conseguimento dello score massimo (30), il docente si riserverà la possibilità di attribuire anche la lode a quegli studenti che dimostreranno di avere acquisito la capacità di elaborare, in assoluta autonomia, efficaci ed efficienti soluzioni a problemi particolarmente complessi. Una valutazione tra 24/30 e 27/30 indica che le capacità progettuali sono buone, ma potrebbero essere migliorate, oppure che la conoscenza delle tematiche dell'insegnamento potrebbe essere più approfondita. Una valutazione inferiore a 24/30 indica che sia capacità progettuali che conoscenza delle tematiche potrebbero essere decisamente migliorati.
OBIETTIVI FORMATIVI	Il corso si propone di introdurre gli studenti ai concetti di base dell'ingegneria del software e allo sviluppo di applicazioni software, utilizzando la metodologia Agile Unified Process (UP), che sfrutta il linguaggio di modellazione UML. Lo studente dovrà saper sviluppare un'applicazione significativa individuando con chiarezza la logica applicativa, l'interazione con le basi di dati e le interfacce richieste dai requisiti. Inoltre dovrà imparare a pianificare il lavoro secondo i canoni dello sviluppo dei progetti: lavoro in team, definizione degli obiettivi e delle fasi di sviluppo.
ORGANIZZAZIONE DELLA DIDATTICA	Il corso ha una forte caratterizzazione progettuale. Per questa ragione è suddiviso in 24 ore di lezione e 48 ore di laboratorio, durante le quali gli studenti avranno l'opportunità di apprendere nuove metodologie ed applicarle in concreto a specifici casi di studio. I casi di studio considerati provengono direttamente dal mondo reale, riflettendo la complessità dei sistemi realizzati nell'ambito di contesti industriali. Le metodologie e gli strumenti di progettazione e sviluppo sono aggiornati rispetto a quanto attualmente utilizzato in ambito lavorativo. Si programmano interazioni sia dirette (attraverso seminari invitati) che indirette (attraverso materiale fornito e/o revisionato) con professionisti dal mondo delle industrie. Durante le lezioni, vengono organizzati dei momenti di brain storming, come anche la presentazione di soluzioni dinnanzi a un pubblico di "peer", composto da altri studenti ed, eventualmente, docenti e professionisti del mondo industriale.
TESTI CONSIGLIATI	Ingegneria del Software, 10/Ed. Con MyLab, Ian Sommerville, Pearson, ISBN: 9788891902245 (2017). Patterns of Enterprise Application Architecture, Martin Fowler, Addison Wesley (2003).

PROGRAMMA

ORE	Lezioni
6	FONDAMENTI: Presentazione del corso. Richiami su Programmazione Orientata agli Oggetti. Sviluppo di software professionale, best practices e versioning. Modelli dei processi software, attività di processo, miglioramento dei processi. Sviluppo software con i metodi agili. Tecniche di sviluppo agile. Project Management Agile: Scrum. Esempi pratici di progettazione secondo metodologia plan-driven (a cascata, usando Gantt) e Agile (Scrum, con storie utente).
6	REQUISITI, ANALISI E PROGETTAZIONE. Tecniche di sviluppo agile su larga scala. Esempi e discussione. Elementi di Ingegneria dei Requisiti. Storie e Scenari Utente. Unified Modeling Language (UML). Casi d'uso. Modelli di sistema e principali diagrammi (caso d'uso, sequenza, stato, classe).
6	DESIGN PATTERN: La teoria dei design pattern. Pattern della "Gang of Four" (GoF). Relazioni tra i design pattern. Catalogo completo dei pattern GoF modellati in UML e implementati in Java. Inversion of Control. Dependency Injection. Architetture software basate su pattern. Sviluppo sistematico di software basato su pattern.
2	TESTING: Progettazione guidata da test. Test di Sviluppo (unità, componenti, integrazione sistema). Test delle Release. Test degli Utenti.
4	SECURITY AND DISTRIBUTED: Open Web Application Security Project (OWASP) e lista delle applicazioni a rischio. Esempi di progettazione non sicura del software. Principi per la progettazione di software sicuro. Secure design pattern. Sistemi distribuiti. Architetture Client-Server, Master-Slave, Peer-to-Peer.

ORE	Laboratori
6	DESIGN PATTERN: Esercizi e implementazione su utilizzo dei design pattern.
6	TESTING: Applicazione pratica su esempi e implementazione. Progettazione guidata da test: JUnit e applicazioni.
6	SECURITY BY DESIGN: Esercizi e applicazioni.
6	PROGETTAZIONE DI UN CASO DI STUDIO: Distinzione tra requisiti funzionali e non funzionali nel caso specifico del progetto assegnato. Disegno logico architetturale del sistema.
6	PROGETTAZIONE DI UN CASO DI STUDIO: Formulazione delle user story (nella forma WHO, WHAT, WHY) per il progetto assegnato. Definizione e dimensionamento dello Sprint. Sprint backlog.
6	PROGETTAZIONE DI UN CASO DI STUDIO: Sprint Poker planning. Progettazione di Web Applications, inclusi framework e tecnologie da utilizzare.
6	PROGETTAZIONE DI UN CASO DI STUDIO: Applicazione della Metodologia Agile: prima si progettano le unità usando i design pattern, poi si scrivono i test in Junit, poi si scrive il codice delle unità; alla fine di ogni Sprint si effettua il test di sistema, che avrà via via funzionalità più articolate.
6	PROGETTAZIONE DI UN CASO DI STUDIO: Proseguimento di progettazione e sviluppo, completando almeno uno sprint. Cambio di specifiche. Refactoring.