



# UNIVERSITÀ DEGLI STUDI DI PALERMO

<b>DIPARTIMENTO</b>	Matematica e Informatica
<b>ANNO ACCADEMICO OFFERTA</b>	2020/2021
<b>ANNO ACCADEMICO EROGAZIONE</b>	2022/2023
<b>CORSO DILAUREA</b>	INFORMATICA
<b>INSEGNAMENTO</b>	INGEGNERIA E SICUREZZA DEL SOFTWARE
<b>TIPO DI ATTIVITA'</b>	B
<b>AMBITO</b>	50166-Discipline Informatiche
<b>CODICE INSEGNAMENTO</b>	20632
<b>SETTORI SCIENTIFICO-DISCIPLINARI</b>	INF/01
<b>DOCENTE RESPONSABILE</b>	ROMBO SIMONA ESTER Professore Ordinario Univ. di PALERMO
<b>ALTRI DOCENTI</b>	
<b>CFU</b>	6
<b>NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE</b>	78
<b>NUMERO DI ORE RISERVATE ALLA DIDATTICA ASSISTITA</b>	72
<b>PROPEDEUTICITA'</b>	
<b>MUTUAZIONI</b>	
<b>ANNO DI CORSO</b>	3
<b>PERIODO DELLE LEZIONI</b>	1° semestre
<b>MODALITA' DI FREQUENZA</b>	Facoltativa
<b>TIPO DI VALUTAZIONE</b>	Voto in trentesimi
<b>ORARIO DI RICEVIMENTO DEGLI STUDENTI</b>	<b>ROMBO SIMONA ESTER</b> 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

<b>PREREQUISITI</b>	Programmazione e programmazione orientata agli oggetti. Progettazione e interrogazione di database.
<b>RISULTATI DI APPRENDIMENTO ATTESI</b>	Lo studente acquisirà: Conoscenze di base e capacità di comprensione sull'ingegneria del software e conoscenza della metodologia Agile e dell'uso di UML in tale ambito. Capacità di applicare metodologie Agili alla progettazione di un applicativo; conoscenza di design pattern di progettazione e architetturali; capacità 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. Capacità di apprendere nuove metodologie per la progettazione e l'implementazione del software.
<b>VALUTAZIONE DELL'APPRENDIMENTO</b>	L'esame è composto da due parti: (1) stesura di un progetto 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 capacità 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. 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.
<b>OBIETTIVI FORMATIVI</b>	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.
<b>ORGANIZZAZIONE DELLA DIDATTICA</b>	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 ed applicare in concreto le metodologie apprese.
<b>TESTI CONSIGLIATI</b>	Ingegneria del Software, Ian Sommerville, Pearson (2017). Patterns of Enterprise Application Architecture, Martin Fowler, Addison Wesley (2003).

## PROGRAMMA

ORE	Lezioni
6	FONDAMENTI: Presentazione del corso. Sviluppo di software professionale. 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. 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.

ORE	Laboratori
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 architetture 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.
6	PROGETTAZIONE DI UN CASO DI STUDIO: Applicazione della Metodologia Agile: prima si progettano le unita' usando i design pattern, poi si scrivono i test in Junit, poi si scrive il codice delle unita'; alla fine di ogni Sprint si effettua il test di sistema, che avra' via via funzionalita' piu' articolate.
6	PROGETTAZIONE DI UN CASO DI STUDIO: Proseguimento di progettazione e sviluppo, completando almeno uno sprint.