



UNIVERSITÀ DEGLI STUDI DI PALERMO

DIPARTIMENTO	Matematica e Informatica
ANNO ACCADEMICO OFFERTA	2017/2018
ANNO ACCADEMICO EROGAZIONE	2019/2020
CORSO DILAUREA	INFORMATICA
INSEGNAMENTO	INGEGNERIA DEL SOFTWARE
TIPO DI ATTIVITA'	B
AMBITO	50166-Discipline Informatiche
CODICE INSEGNAMENTO	03968
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	102
NUMERO DI ORE RISERVATE ALLA DIDATTICA ASSISTITA	48
PROPEDEUTICITA'	05880 - PROGRAMMAZIONE E LABORATORIO C.I. 16670 - ALGORITMI E STRUTTURE DATI 16784 - SISTEMI OPERATIVI 16450 - ARCHITETTURE DEGLI ELABORATORI 16671 - INFORMATICA TEORICA
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	Algoritmi e Strutture Dati, Basi di Dati, Linguaggi di Programmazione ad oggetti (Java)
RISULTATI DI APPRENDIMENTO ATTESI	<p>Conoscenza e capacita' di comprensione</p> <p>Acquisizione dei concetti e delle tecniche fondamentali per la raccolta dei requisiti, stesura delle specifiche, pianificazione, progettazione, implementazione, integrazione e manutenzione di un prodotto software. Conoscenza del linguaggio UML (Unified Modeling Language).</p> <p>Conoscenza degli aspetti fondamentali del processo di sviluppo del software (con particolare riferimento al software object-oriented)</p> <p>Capacita' di applicare conoscenza e comprensione</p> <p>Capacita' di raccogliere e formalizzare i requisiti del sistema. Capacita' di progettare un sistema software anche complesso. Capacita' di stimare costi e tempi. Capacita' di dimensionare correttamente i vari componenti. Capacita' di utilizzo dell'UML come linguaggio di modellazione. Utilizzo degli strumenti di ausilio alla progettazione piu' comuni.</p> <p>Autonomia di giudizio</p> <p>Saper identificare i requisiti. Saper formalizzare i requisiti. Saper individuare, a diversi livelli di astrazione, le soluzioni necessarie al raggiungimento degli obiettivi funzionali del sistema. Saper individuare le architetture piu' idonee per soddisfare i requisiti funzionali e non. Saper modellare un sistema informatico dai requisiti fino alla implementazione, test e configurazione/installazione. Saper attuare scelte risk-driven nella pianificazione e attuazione del progetto.</p> <p>Abilita' comunicative</p> <p>Proprieta' di espressione nella presentazione delle scelte progettuali e delle motivazioni che hanno portato ad esse.</p> <p>Capacita' d'apprendimento</p> <p>Sapere affrontare la decomposizione di un problema, identificare i requisiti e le soluzioni tecnologiche necessarie. Sapere contestualizzare le abilita' acquisite in problemi concreti nell'ambito lavorativo.</p>
VALUTAZIONE DELL'APPRENDIMENTO	<p>La valutazione si basera' su un progetto svolto in gruppo e su un colloquio. I gruppi saranno composti da 3 persone. Ogni gruppo potra' svolgere un progetto di propria scelta ma dovra' proporre una descrizione breve del problema al docente per l'approvazione.</p> <p>Il progetto sara' composto dai documenti di analisi dei requisiti, progetto del sistema, progetto degli oggetti e piano di gestione del progetto (project management plan). I documenti andranno redatti secondo dei modelli presentati a lezione.</p> <p>Il progetto verra' valutato considerando alcuni aspetti fondamentali:</p> <ol style="list-style-type: none">1) uso corretto dell'UML per ognuno dei tipi di diagrammi richiesti2) applicazione corretta dei concetti di progettazione studiati3) completezza/correttezza della documentazione4) tracciabilita' dei requisiti e logica consequenzialita' delle scelte fatte a partire dai requisiti iniziali e dalle scelte precedenti. <p>Particolare enfasi verra' data al rispetto della formalita' dei documenti di progetto. I requisiti per conseguire il punteggio minimo per accedere alla seconda parte dell'esame consistono nella capacita' di realizzare un progetto che, seppure in modo basilare, rispetti le specifiche assegnate e sia formalmente corretto (anche nell'uso del linguaggio UML).</p> <p>La valutazione del progetto avvera' tramite una apposita scheda di valutazione che sara' resa disponibile agli studenti tramite il sito web del corso.</p> <p>Il colloquio servira' per verificare le conoscenze acquisite durante il corso, l'autonomia nell'approfondire contenuti anche complessi e la capacita' individuale di trovare soluzioni a problemi proposti. La prova orale avra' inizio con la discussione del progetto svolto. Quindi, verra' verificata la capacita' di analisi/sintesi e l'autonomia di giudizio dello studente attraverso un numero adeguato di domande. La valutazione della prova orale andra' a integrare il punteggio conseguito da ciascuno studente a seguito della valutazione del progetto.</p>
OBIETTIVI FORMATIVI	<p>Il modulo si propone di fornire allo studente le conoscenze e competenze necessarie per affrontare la progettazione di un sistema informatico. L'obiettivo principale del corso e' lo studio del processo di sviluppo del software (con particolare riferimento ai software object-oriented). Verranno esaminate le tecniche di raccolta dei requisiti, stesura delle specifiche, pianificazione, progettazione, implementazione, testing, integrazione e manutenzione.</p>
ORGANIZZAZIONE DELLA DIDATTICA	Lezioni frontali, esercitazioni in aula, attivita' in laboratorio

TESTI CONSIGLIATI	<p>Testo di riferimento Ian Sommerville Ingegneria del software 8/Ed. 2007 pp. 848 ISBN 9788871923543</p> <p>Testo di approfondimento B. Bruegge, A. H. Dutoit. Object-Oriented Software Engineering: Using UML, Patterns and Java. Third edition. International edition. ISBN: 0-13-606125-7. Prentice Hall</p>
--------------------------	--

PROGRAMMA

ORE	Lezioni
2	Introduzione all'Ingegneria del Software. Concetti di: progetto, attività, risorsa, task, workproduct, sistema, modello, documento, obiettivi (goal), requisiti, vincoli, notazioni, metodi e metodologie.
2	Le fasi principali dello sviluppo: raccolta requisiti, analisi dei requisiti, progetto di sistema, progetto esecutivo o degli oggetti, implementazione, gestione del progetto, testing, ciclo di vita del software.
3	Introduzione all'UML, diagrammi dei casi d'uso, diagrammi delle classi, diagrammi di sequenza e collaborazione. Diagrammi di stato diagrammi di attività, diagrammi di dislocazione, organizzazione dei diagrammi, estensione dei diagrammi, concetti di sistema, modello e vista. Modellazione object-oriented.
4	Introduzione alla raccolta dei requisiti. Concetti fondamentali: requisiti funzionali, requisiti non funzionali e pseudo-requisiti, livelli di descrizione. Principali attributi delle specifiche (correttezza, completezza, ...). Classificazione delle attività di raccolta dei requisiti.
6	Analisi dei requisiti: Identificazione degli attori, degli scenari, dei casi d'uso, dei casi d'uso, delle relazioni tra attori e casi d'uso. Identificazione degli oggetti d'analisi, identificazione dei requisiti non funzionali. Modelli di analisi: funzionale, degli oggetti, dinamico. Concetti di analisi: oggetti di tipo entità, di confine e di controllo. Passaggio dai casi d'uso agli oggetti, identificazione degli oggetti entità, di confine, di controllo, modellazione delle interazioni, identificazione delle associazioni e degli attributi, modellazione del comportamento degli oggetti.
5	Progettazione di sistema. Introduzione, concetti e attività principali. Architetture software: client-server peer to peer, pipe and filter. Attività della progettazione di sistema. Identificazione dei sottosistemi. La mappatura dei sottosistemi su processori e componenti. La definizione dei depositi di dati. La definizione dal controllo l'accesso. La progettazione del flusso di controllo: procedure-driven, event-driven, threads. La progettazione delle condizioni di confine.
5	Progettazione di dettaglio. Introduzione, concetti della progettazione del modello ad oggetti. Attività della progettazione del modello oggetti; oggetti d'analisi e oggetti della soluzione. Tipi, signature, visibilità. Contratti: invarianti, pre-condizioni, post-condizioni. Object Constraint Language. Identificazione degli attributi e delle operazioni mancanti; specifica di tipi, signature e visibilità. Specifica dei vincoli (constraint), delle eccezioni; identificazione e adattamento delle librerie di classi; realizzazione delle associazioni. Incrementare il riuso.
4	Testing. Introduzione al testing; tecniche di controllo della qualità; tecniche per evitare guasti; tecniche per la scoperta dei guasti tecniche per tollerare i guasti. Concetti di test: componente, guasto, errore, malfunzionamento, test case, test stub/driver, correzione. Attività di testing: ispezione dei componenti; unit testing; test di integrazione; test di sistema. - test di integrazione: strategie del test di integrazione (big bang, bottom-up, top-down, sandwich, modified sandwich). - test di sistema: test funzionale, test di performance, test pilota (alpha test, beta test), test di accettazione e test di installazione. Pianificazione del test; documentazione del test.
4	Project Management. Elementi fondamentali di project management Caratteristiche fondamentali del progetto Attività (ordinarie, di riepilogo, cardine). Struttura delle attività. Relazioni tra attività. Le risorse (il calendario, i costi). I costi fissi del progetto. Le relazioni di progetto. Il progetto iniziale. Il progetto con previsioni. Il progetto con variazioni.

PROGRAMMA

ORE	Lezioni
13	Utilizzo di CASE tool per la modellazione UML. Esempi ed esercizi di: <ul style="list-style-type: none">- raccolta e documentazione dei requisiti.- analisi e documentazione dei requisiti.- definizione e documentazione della progettazione di sistema.- definizione e documentazione della progettazione di dettaglio.- definizione, pianificazione e documentazione dei casi di test- project management