



UNIVERSITÀ DEGLI STUDI DI PALERMO

DIPARTIMENTO	Ingegneria
ANNO ACCADEMICO OFFERTA	2021/2022
ANNO ACCADEMICO EROGAZIONE	2023/2024
CORSO DILAUREA	INGEGNERIA INFORMATICA
INSEGNAMENTO	INGEGNERIA DEL SOFTWARE
TIPO DI ATTIVITA'	C
AMBITO	10655-Attività formative affini o integrative
CODICE INSEGNAMENTO	03968
SETTORI SCIENTIFICO-DISCIPLINARI	ING-INF/05
DOCENTE RESPONSABILE	SEIDITA VALERIA Ricercatore Univ. di PALERMO
ALTRI DOCENTI	
CFU	9
NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE	144
NUMERO DI ORE RISERVATE ALLA DIDATTICA ASSISTITA	81
PROPEDEUTICITA'	
MUTUAZIONI	
ANNO DI CORSO	3
PERIODO DELLE LEZIONI	2° semestre
MODALITA' DI FREQUENZA	Facoltativa
TIPO DI VALUTAZIONE	Voto in trentesimi
ORARIO DI RICEVIMENTO DEGLI STUDENTI	SEIDITA VALERIA Giovedì 11:00 13:00 Fino alla fine della attuale situazione dovuta al COVID ed alla ripresa delle normali attività, il ricevimento si svolgerà attraverso la piattaforma Microsoft Teams, previa prenotazione. CODICE TEAMS: jb05vcs (canale Ricevimento).

PREREQUISITI	<p>Programmazione C e Java</p> <p>Algoritmi e strutture dati</p> <p>Basi di dati</p>
RISULTATI DI APPRENDIMENTO ATTESI	<p>Risultati attesi in accordo con i descrittori di Dublino:</p> <p>Obiettivo 1 - Conoscenza e capacita' di comprensione Lo studente acquisira' i concetti e le tecniche fondamentali per la raccolta dei requisiti, stesura delle specifiche, pianificazione, progettazione, implementazione, integrazione e manutenzione di un prodotto software. Acquisira' anche una buona conoscenza del linguaggio UML (Unified Modeling Language) e degli aspetti fondamentali del processo di sviluppo del software (con particolare riferimento al software object-oriented). Per il raggiungimento di questo obiettivo il corso comprende lezioni frontali e analisi e discussione di casi di studio.</p> <p>Obiettivo 2 - Capacità di applicare conoscenza e comprensione Lo studente acquisira' la capacita' di raccogliere e formalizzare i requisiti del sistema, di progettare un sistema software anche complesso e di stimare costi e tempi. Inoltre lo studente sara' in grado di dimensionare correttamente i vari componenti di un sistema software e di utilizzare UML come linguaggio di modellazione insieme con gli strumenti di ausilio alla progettazione piu' comuni. Per il raggiungimento di questo obiettivo, il corso comprende analisi e discussioni su casi di studio, esercitazioni di gruppo orientate all'implementazione di metodologie per lo sviluppo di software con l'uso di idonei ambienti di sviluppo e la preparazione di un progetto svolto autonomamente.</p> <p>Obiettivo 3 - Autonomia di giudizio Lo studente acquisira' le metodologie di progettazione, implementazione e valutazione di architetture di sistemi informatici e analizzera' diversi casi di studio. Sara' quindi in grado di analizzare i dati, anche limitati e incompleti, a sua disposizione e proporre soluzioni adeguate per problemi nuovi integrando le conoscenze acquisite durante il corso. Sara' in grado di analizzare pregi e difetti delle soluzioni proposte, attuare scelte risk-driven nella pianificazione e attuazione del progetto e modellare un sistema informatico dai requisiti fino alla implementazione, test e configurazione/installazione. Per il raggiungimento di questo obiettivo il corso comprende: analisi e discussioni su casi di studio; lezioni ed esercitazioni sulla progettazione, implementazione e valutazione di architetture software; preparazione di un progetto svolto autonomamente riguardante la progettazione e la realizzazione di un sistema software.</p> <p>Obiettivo 4 - Abilita' comunicative Lo studente acquisira' la capacita' di comunicare con competenza e proprieta' di linguaggio le conoscenze acquisite riguardanti le problematiche relative alla progettazione, implementazione e valutazione di sistemi informatici. Per il raggiungimento di questo obiettivo, il corso comprende esercitazioni sulla progettazione e implementazione di sistemi software e presentazioni e discussioni in aula di progetti e implementazioni.</p> <p>Obiettivo 5 - Capacita' d'apprendimento Lo studente sara' in grado di apprendere in autonomia le problematiche specifiche relative alla progettazione e realizzazione del software. Per il raggiungimento di questo obiettivo, il corso comprende esercitazioni riguardanti l'implementazione di sistemi software e presentazioni in aula di progetti e implementazioni</p>
VALUTAZIONE DELL'APPRENDIMENTO	<p>La valutazione dell'apprendimento sara' focalizzata sulla valutazione dei risultati attesi (si veda sotto) in accordo con i descrittori di Dublino. Il voto finale sara' dato in trentesimi e variera' da 18/30 a 30/30 con lode.</p> <ul style="list-style-type: none"> - Valutazione dell'obiettivo 1: Conoscenza e capacita' di comprensione L'obiettivo sara' valutato attraverso la discussione orale sugli argomenti del programma e su un progetto preparato in gruppo o autonomamente riguardante la realizzazione di un sistema software. - Valutazione dell'obiettivo 2: Capacita' di applicare conoscenza e comprensione L'obiettivo sara' valutato attraverso la discussione degli elaborati preparati durante le esercitazioni e la discussione del progetto preparato autonomamente. - Valutazione dell'obiettivo 3: autonomia di giudizio L'obiettivo sara' valutato attraverso la discussione critica delle tematiche teoriche svolte durante il corso e la discussione del progetto preparato autonomamente (o in gruppo) riguardante la progettazione e la realizzazione di un sistema software. - Valutazione dell'obiettivo 4: abilita' comunicative L'obiettivo sara' valutato attraverso un colloquio orale sugli argomenti del corso, la discussione su casi di studio, la presentazione di un progetto preparato autonomamente riguardante la progettazione e la realizzazione di un sistema software. - Valutazione dell'obiettivo 5: Capacita' di apprendimento

	<p>L'obiettivo sara' valutato attraverso una prova orale relativa alla progettazione e realizzazione del software, nonche' la discussione di un progetto preparato autonomamente riguardante la progettazione e la realizzazione di un sistema software.</p> <p>Per superare l'esame lo studente dovra' ottenere una valutazione positiva su tutti gli obiettivi. Il voto terra' conto sia della valutazione del progetto che della esposizione ed illustrazione orale dello stesso. Il progetto verra' valutato con un voto sufficiente (18/30) se risponde almeno in modo minimale alle specifiche date mentre verra' valutato eccellente (30/30 e lode) se lo studente avra' elaborato autonomamente le specifiche ottenendo cosi' un risultato personalizzato con una perfetta tracciabilita' dai requisiti al codice. La discussione orale verra' valutata con un voto sufficiente se lo studente dimostrera' piena padronanza di almeno una della fasi della progettazione mentre verra' valutato eccellente se dimostrera' piena padronanza di tutte le fasi della progettazione del software richiesto. La prova orale verra' effettuata solo se al progetto sara' stata assegnata una valutazione superiore al 18. Il voto finale verra' calcolato come media tra il voto assegnato al progetto e quello assegnato all'orale.</p>
OBIETTIVI FORMATIVI	<p>Il corso implementa gli obiettivi formativi previsti dal Corso di Laurea Triennale in Ingegneria Informaticaper quanto riguarda la progettazione del software. In accordo con gli obiettivi formativi qualificanti della classe Ingegneria dell'Informazione.</p> <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	<p>Il corso e' articolato in: lezioni frontali (62 ore) esercitazioni (19 ore)</p>
TESTI CONSIGLIATI	<p>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> <p>Ian Sommerville Ingegneria del software 8/Ed. 2007 pp. 848 ISBN 9788871923543</p> <p>M. Fowler. UML Distilled. Pearson. ISBN 0-321-19368-7</p> <p>Jim Arlow, Ila Neustadt. UML2 e Unified Process - analisi e progettazione Object Oriented, Addison-Wesley- ISBN 978-0-321-32127-5</p> <p>Craig Larman. Applicare UML e i pattern. Pearson. ISBN 9788891904591</p>

PROGRAMMA

ORE	Lezioni
2	Introduzione all'Ingegneria del Software. Concetti di: progetto, attivita, 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.
6	Introduzione all'UML, diagrammi dei casi d'uso, diagrammi delle classi, diagrammi di sequenza e collaborazione. Diagrammi di stato diagrammi di attivita, diagrammi di dislocazione, organizzazione dei diagrammi, estensione dei diagrammi, concetti di sistema, modello e vista. Modellazione object-oriented.
8	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 attivita' 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 entita, di confine e di controllo. Passaggio dai casi d'uso agli oggetti, identificazione degli oggetti entita, di confine, di controllo, modellazione delle interazioni, identificazione delle associazioni e degli attributi, modellazione del comportamento degli oggetti.
6	Progettazione di sistema. Introduzione, concetti e attivita' principali. Architetture software: client-server peer to peer, pipe and filter. Attivita' 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.System Desig

PROGRAMMA

ORE	Lezioni
3	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.
3	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.
2	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.
6	Design Pattern
6	Unified Process
12	Ingegneria del Software avanzata: Component-based, service-oriented, aspect oriented.
ORE	Esercitazioni
12	Utilizzo di CASE tool per la modellazione UML. Esempi ed esercizi di: - 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
3	Utilizzo pratico di UP e dei pattern di design
2	Project management
2	Testing