



# UNIVERSITÀ DEGLI STUDI DI PALERMO

<b>DIPARTIMENTO</b>	Ingegneria
<b>ANNO ACCADEMICO OFFERTA</b>	2016/2017
<b>ANNO ACCADEMICO EROGAZIONE</b>	2018/2019
<b>CORSO DILAUREA</b>	INGEGNERIA GESTIONALE E INFORMATICA
<b>INSEGNAMENTO</b>	BASI DI DATI E PROGETTAZIONE DEL SOFTWARE
<b>TIPO DI ATTIVITA'</b>	B
<b>AMBITO</b>	50289-Ingegneria informatica
<b>CODICE INSEGNAMENTO</b>	15979
<b>SETTORI SCIENTIFICO-DISCIPLINARI</b>	ING-INF/05
<b>DOCENTE RESPONSABILE</b>	SEIDITA VALERIA      Professore Associato      Univ. di PALERMO
<b>ALTRI DOCENTI</b>	
<b>CFU</b>	9
<b>NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE</b>	144
<b>NUMERO DI ORE RISERVATE ALLA DIDATTICA ASSISTITA</b>	81
<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>SEIDITA VALERIA</b> Lunedì    14:30    15:30    Team dei ricevimenti del docente (codice: ce0vzsz).

<p><b>PREREQUISITI</b></p>	<p>Programmazione C e Java Algoritmi e strutture dati</p>
<p><b>RISULTATI DI APPRENDIMENTO ATTESI</b></p>	<p>Risultati attesi in accordo con i descrittori di Dublino:  <b>Obiettivo 1 - Conoscenza e capacita' di comprensione</b>                  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). Inoltre lo studente acquisira' le conoscenze teoriche relative alle basi di dati e alla progettazione di sistemi di tipo client-server. Per il raggiungimento di questo obiettivo il corso comprende lezioni frontali e analisi e discussione di casi di studio.  <b>Obiettivo 2 - Conoscenza e capacita' di comprensione</b>                  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 (una applicazione client-server con interfacciamento con un database) svolto autonomamente.  <b>Obiettivo 3 - autonomia di giudizio</b>                  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 propporre 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 (una applicazione client-server con interfacciamento con un database).  <b>Obiettivo 4 - Abilita' comunicative</b>                  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.  <b>Obiettivo 5 - Capacita' d'apprendimento</b>                  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>
<p><b>VALUTAZIONE DELL'APPRENDIMENTO</b></p>	<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.                  - 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                  L'obiettivo sara' valutato attraverso una prova orale relativa alla progettazione e</p>

	realizzazione del software, nonché la discussione di un progetto preparato autonomamente riguardante la progettazione e la realizzazione di un sistema software.
<b>OBIETTIVI FORMATIVI</b>	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 è 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. Inoltre verrà dato ampio spazio alla progettazione concettuale e logica delle basi di dati.
<b>ORGANIZZAZIONE DELLA DIDATTICA</b>	Il corso è articolato in: lezioni frontali (49 ore) discussioni di casi studio (17 ore) laboratorio ed esercitazioni (15 ore)
<b>TESTI CONSIGLIATI</b>	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 Ian Sommerville Ingegneria del software 8/Ed. 2007 pp. 848 ISBN 9788871923543 M. Fowler. UML Distilled. Pearson Jim Arlow, Ila Neustadt. UML2 e Unified Process - analisi e progettazione Object Oriented, Addison-Wesley P. Atzeni, S. Ceri, S. Paraboschi, R. Torlone. Basi di Dati – Modelli e Linguaggi di Interrogazione. McGraw-Hill

### 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.
6	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.
6	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, 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 del controllo dell'accesso. La progettazione del flusso di controllo: procedure-driven, event-driven, threads. La progettazione delle condizioni di confine. System Desig
4	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.
3	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.
3	Introduzione alle basi di dati relazionali
3	Progettazione concettuale di basi di dati con il modello ER
2	Progettazione logica di basi di dati
2	SQL
2	Accesso a basi di dati in ambiente Java

<b>ORE</b>	<b>Esercitazioni</b>
5	Esempi di raccolta, analisi e documentazione dei requisiti. Esempi di definizione e documentazione della progettazione di sistema e di dettaglio. Esempi definizione, pianificazione e documentazione dei casi di test. Esempi di project management
12	Confronto di ambienti per basi di dati relazionali Esempi di uso del linguaggio SQL Discussione sull'amministrazione di basi di dati Esempi di progettazione OO e programmazione in Java Discussione sulla realizzazione di interfacce utente Esempi di progettazione integrata di applicazioni client-server
<b>ORE</b>	<b>Laboratori</b>
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	Ambienti per basi di dati relazionali Linguaggio SQL Progettazione e amministrazione di basi di dati