



Insegnamento	Ingegneria del Software
Livello e corso di studio	Laurea Magistrale in Ingegneria Informatica e dell'Automazione (LM32)
Settore scientifico disciplinare (SSD)	ING-INF/05 - Sistemi di elaborazione delle informazioni
Anno di corso	1
Numero totale di crediti	9
Propedeuticità	-
Docente	Antonino Longo Minnolo <a href="https://ricerca.unicusano.it/author/antonino-longo/">https://ricerca.unicusano.it/author/antonino-longo/</a> Nickname: antonino-longo <a href="mailto:antonino.longo@unicusano.it">antonino.longo@unicusano.it</a> Orario di ricevimento: da concordare previo appuntamento con il docente
Presentazione	Il Corso di Ingegneria del Software ha lo scopo di far acquisire allo studente una buona conoscenza dei principi che governano la progettazione del software. Il Corso propone i concetti basilari relativamente al processo di sviluppo del software, con particolare riferimento ai software orientati agli oggetti. Inoltre, obiettivo formativo del Corso è fornire allo studente una conoscenza nel dettaglio sulle tecniche di raccolta dei requisiti, stesura delle specifiche, pianificazione, progettazione, implementazione, testing, integrazione e manutenzione del software. Le Etivity associate al Corso sviluppano le competenze necessarie a progettare e sviluppare prodotti software attraverso l'uso di idonei ambienti di ingegnerizzazione del software.
Obiettivi formativi	Il Corso ha i seguenti obiettivi formativi: <ol style="list-style-type: none"><li>1. Illustrare le principali tecniche di raccolta dei requisiti</li><li>2. Illustrare le principali metodologie di stesura delle specifiche</li><li>3. Illustrare i principali paradigmi di pianificazione e progettazione del software</li><li>4. Illustrare tecniche di implementazione, testing, integrazione e manutenzione di un progetto software</li><li>5. Affrontare l'implementazione di un software mediante il linguaggio Java</li></ol>
Prerequisiti	La frequenza al Corso richiede la conoscenza dei concetti fondamentali di programmazione e database come apprese nel Corso di Programmazione ad Oggetti (L8) e nel Corso di Basi di Dati (L8), rispettivamente
Risultati di apprendimento attesi	<b>Conoscenza e capacità di comprensione (KNOWLEDGE AND UNDERSTANDING)</b>



Lo studente al termine del Corso avrà conoscenza dei concetti e delle tecniche fondamentali per la raccolta dei requisiti, stesura delle specifiche, pianificazione, progettazione, implementazione, integrazione e manutenzione di un prodotto software. Inoltre, lo studente acquisirà 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, tramite le Etivity gli studenti acquisiranno la capacità di affrontare l'analisi e la discussione di casi di studio concreti.

### **Applicazione delle conoscenze (APPLYING KNOWLEDGE AND UNDERSTANDING)**

Lo studente sarà in grado di raccogliere e formalizzare i requisiti del sistema, di progettare un sistema software anche complesso e di stimare costi e tempi; sarà inoltre 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 più comuni.

Le Etivity prevedono l'applicazione delle conoscenze teoriche a problemi pratici, come analisi e discussioni su casi di studio, ed esercitazioni di gruppo orientate all'implementazione di metodologie per lo sviluppo di software, da svolgere con l'uso di ambienti di sviluppo idonei e la preparazione di un progetto svolto autonomamente.

### **Capacità di trarre conclusioni (ABILITY TO DRAW CONCLUSIONS)**

Lo studente sarà in grado di individuare le corrette metodologie di progettazione, implementazione e valutazione di architetture di sistemi informatici, analizzando diversi casi di studio; sarà inoltre 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. Infine, lo studente sarà in grado di effettuare ricerche bibliografiche, di analizzare ed interpretare le fonti rilevanti, al fine 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.

### **Abilità comunicative (COMMUNICATION SKILLS)**

Lo studente sarà in grado di descrivere e sostenere conversazioni su tematiche relative alla progettazione e implementazione e valutazione del prodotto software, e alla risoluzione dei tipici problemi di gestione del suo ciclo di vita, adoperando una terminologia adeguata.

### **Capacità di apprendere (LEARNING SKILLS)**

Lo studente al termine del Corso sarà in grado di apprendere in autonomia le problematiche specifiche relative alla progettazione e realizzazione del software.



## Organizzazione dell'insegnamento

Il corso è sviluppato attraverso le **lezioni preregistrate audio-video** che compongono, insieme a slide e dispense, i materiali di studio disponibili in piattaforma.

Sono poi proposti dei **test di autovalutazione**, di tipo asincrono, che corredano le lezioni preregistrate e consentono agli studenti di accertare sia la comprensione, sia il grado di conoscenza acquisita dei contenuti di ognuna delle lezioni.

La **didattica interattiva** è svolta nel forum della "classe virtuale" e comprende 4 **Etivity**.

In particolare, il Corso di Ingegneria del Software prevede 9 Crediti formativi. Il carico totale di studio per questo modulo di insegnamento è compreso tra 220 e 250 ore così suddivise in:

- **circa 178,5 ore** per la visualizzazione e lo studio del materiale videoregistrato (22 Ore videoregistrate di Teoria e 5 ore di esercitazioni);
- **circa 50 ore di Didattica Interattiva** per l'elaborazione e la consegna di 4 Etivity;
- **circa 5 ore di Didattica Interattiva** per l'esecuzione dei test di autovalutazione.

Si consiglia di distribuire lo studio della materia uniformemente in un periodo di 9 settimane dedicando tra le 20 e le 30 ore di studio a settimana

## Contenuti del corso

**Modulo 1 – Concetti e definizioni di base dell'Ingegneria del Software (8 lezioni di teoria videoregistrate per un impegno di 21 ore – settimana 1)** dove sono affrontati i seguenti argomenti: Introduzione all'Ingegneria del Software, Concetti di progetto, attività, risorsa, task, work product, sistema, modello, documento, obiettivi (goal), requisiti, vincoli, notazioni, metodi e metodologie. Definizioni di base: prodotti software, caratteristiche generali dei prodotti software. Le qualità del software. 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. I processi per lo sviluppo del software: modello a cascata, sviluppo incrementale; modello iterativo/evolutivo, modello prototipale, modello a spirale, modello agile.

**Modulo 2 – Analisi e specifica dei requisiti (4 lezioni di teoria videoregistrate per un impegno di 14 ore – settimana 1 e 2)** dove sono affrontati i seguenti argomenti: 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. 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. Esercitazione di analisi e documentazione dei requisiti.

**Etivity 1 (15 ore di carico di studio – settimana 2-3) – Esempi di analisi su casi di studio.**



**Modulo 3 – Progettazione del Sistema (5 lezioni di teoria videoregistrate per un impegno di 17,5 ore – settimana 3)** dove sono affrontati i seguenti argomenti: Progettazione di sistema. Introduzione, concetti e attività principali. Progetto del software. Metodi di progetto: approccio top-down, bottom-up, metodi strutturati, strategie funzionali e object oriented. Documentazione del progetto. Parametri di qualità del progetto: coesione, accoppiamento. Progetto dell'architettura logica. Principi di analisi e progettazione orientata agli oggetti: Classi, Oggetti, Overloading, Information hiding, Ereditarietà, Classi astratte, Interfacce, Binding dinamico, Eccezioni. Esercitazione per la definizione e documentazione della progettazione di sistema.

**Modulo 4 – La modellazione e progettazione del software con UML (10 lezioni di teoria videoregistrate per un impegno di 35 ore e 1 lezione di esercitazione per un impegno di 2,5 ore – settimana 3-4-5)**

Generalità su UML (Unified Modeling Language). UML e ciclo di vita. Modellare i requisiti con i casi d'uso. Diagrammi delle classi e degli oggetti. Diagrammi di sequenza. Diagramma degli stati. Diagramma di attività. Diagramma dei componenti e di deployment. Modellazione object-oriented.

**Etivity 2 (15 ore di carico di studio – settimana 3-4) – Utilizzo di tool CASE per la modellazione UML.**

**Modulo 5 – Introduzione al linguaggio Java (7 lezioni di teoria videoregistrate per un impegno di 23 ore – settimana 6)**

Introduzione al linguaggio Java: JVM e JDK. Fondamenti di Object Oriented Programming. L'ambiente di sviluppo ECLIPSE. Elementi caratteristici del linguaggio Java. Le librerie standard di Java. Gestione delle eccezioni. - Manipolazione degli errori. Documentazione con JavaDoc. I/O in Java. La gestione dei file. La programmazione Multithread. Interazione con il File System. Networking in Java. Collegamento ai database: JDBC. Eseguire unit test con JUnit

**Modulo 6 – Verifica, validazione e testing (4 lezioni di teoria videoregistrate per un impegno di 14 ore – settimana 7)**

Il controllo di qualità dei prodotti software: la verifica e la validazione. Introduzione al testing; tecniche di controllo della qualità; tecniche di prevenzione guasti; tecniche per il riconoscimento guasti; tecniche di tolleranza ai guasti. Concetti di test: componente, guasto, errore, malfunzionamento, test case, stub/driver di test, correzione. Attività di testing: ispezione dei componenti; unit testing; test di integrazione e di sistema. Pianificazione dei test, documentazione dei test. Esercitazione su definizione, pianificazione e documentazione dei casi di test. Eseguire unit test con JUnit.



	<p><b>Etivity 3 (15 ore di carico di studio – settimana 7) – Progettazione e implementazione di un sistema completo in Java.</b></p> <p><b>Modulo 7 – Gestione del progetto e Design Patterns (8 lezioni di teoria videoregistrate per un impegno di 28 ore – settimana 8)</b> dove sono affrontati i seguenti argomenti: 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. Design Pattern. Esercitazione sul project management.</p> <p><b>Modulo 8 – Progettazione Agile (3 lezioni di teoria videoregistrate per un impegno di 10,5 ore – settimana 9)</b> dove sono affrontati i seguenti argomenti: I principi agili. Metodi agili. Metodologia Scrum. Scrum Team: il Product Owner, il Team di Sviluppo, lo Scrum Master. Gli Eventi di Scrum: lo Sprint, Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective. .Gli Artefatti di Scrum: Product Backlog, Sprint Backlog, Incremento. Gestire i progetti con Scrum.</p> <p><b>Modulo 9 – Software development management (3 lezioni di teoria videoregistrate per un impegno di 10,5 ore e 1 lezione di esercitazione per un impegno di 2,5 ore – settimana 9)</b> dove sono affrontati i seguenti argomenti: Software configuration management. Configuration item, version, configurazioni, repository. Utilizzo di tool di versioning (CVS, SVN, Git). Gestione delle build, release e branch.</p> <p><b>Esercitazioni su compiti d'esame (2 lezioni di esercitazione per un impegno di 5 ore)</b></p>
Materiali di studio	<p>MATERIALI DIDATTICI A CURA DEL DOCENTE</p> <p>Il materiale didattico presente in piattaforma è suddiviso in 9 moduli. Essi ricoprono interamente il programma e ciascuno di essi contiene dispense, slide e videolezioni in cui il docente commenta le slide. Tale materiale contiene tutti gli elementi necessari per affrontare lo studio della materia.</p> <p>Testi consigliati:</p> <ul style="list-style-type: none"><li>• Sommerville Ingegneria del software 8/Ed. 2007 pp. 848 ISBN 9788871923543</li><li>• J. Arlow, I. Neustadt. UML2 e Unified Process - analisi e progettazione Object Oriented, Addison-Wesley</li><li>• C. De Sio, Cesari (2014) - Manuale di Java 8. Programmazione orientata agli oggetti con Java standard edition 8 – 1° Edizione - Hoepli – ISBN 8820362910</li><li>• Herbert Schildt – Java la guida completa – McGrawill – ISBN 9788838667664</li></ul>
Modalità di verifica dell'apprendimento	<p>L'esame consiste nello svolgimento di una prova scritta tendente ad accertare le capacità di analisi e rielaborazione dei concetti acquisiti.</p> <p>In accordo con il modello formativo del Corso di Studi, La valutazione finale dell'insegnamento, espressa in trentesimi, prende in considerazione anche l'attività svolta</p>



	<p>in itinere dallo studente e valutata attraverso il punteggio assegnato alle quattro E-tivity proposte (da 0 a 6).</p> <p>La prova scritta se svolta per intero (si veda più avanti) prevede, di norma, due esercizi per accertare le capacità di analisi e rielaborazione dei concetti acquisiti ed una domanda di teoria.</p> <p>I risultati di apprendimento attesi circa le conoscenze della materia e la capacità di applicarle sono valutate dalla prova scritta, mentre le abilità comunicative, la capacità di trarre conclusioni e la capacità di autoapprendimento sono valutate in itinere attraverso le E-tivity.</p> <p>È possibile effettuare l'esame attraverso due prove parziali scritte</p> <ul style="list-style-type: none"><li>✓ Parte - 1 moduli da 1 a 4;</li><li>✓ Parte - 2 moduli da 5 a 9.</li></ul> <p>In sede di valutazione finale, si terrà conto anche della proficua partecipazione ai forum (aule virtuali) e al corretto svolgimento delle e-tivity proposte.</p>
Criteri per l'assegnazione dell'elaborato finale	L'assegnazione dell'elaborato finale avverrà sulla base di un colloquio con il docente in cui lo studente manifesterà i propri specifici interessi in relazione a qualche argomento che intende approfondire; non esistono preclusioni alla richiesta di assegnazione della tesi e non è prevista una media particolare per poterla richiedere.