



Principali informazioni sull'insegnamento

Denominazione dell'insegnamento	Ingegneria del Software	
Corso di studio	Informatica	
Anno Accademico	2024/25	
Crediti formativi universitari (CFU) / European Credit Transfer and Accumulation System (ECTS)	9 CFU	
Settore Scientifico Disciplinare	INF/01	
Lingua di erogazione	Italiano	
Anno di corso	Secondo	
Periodo di erogazione	2^ semestre, le date esatte sono riportate nel manifesto/regolamento	
Obbligo di frequenza	La frequenza è fortemente raccomandata	
Sito web del corso di studio	https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/informatica-270/laurea-triennale-in-informatica-d.m.-270-1	

Docente/i	
Nome e cognome	Filippo Lanubile
Indirizzo mail	filippo.lanubile@uniba.it
Telefono	via MS Teams
Sede	Dipartimento di Informatica, Via Orabona 4, 70125, Bari. 6^ piano.
Sede virtuale	Piattaforma e-Learning UNIBA - https://elearning.uniba.it/
Sito web del docente	https://collab.di.uniba.it/lanubile/
Ricevimento (giorni, orari e modalità, es. su appuntamento)	su appuntamento



Syllabus	
Obiettivi formativi	L'insegnamento si propone di fornire agli studenti una generale comprensione della disciplina tecnologica e manageriale riguardante lo sviluppo del software che (1) deve essere modificabile per tutta la durata di vita prevista, (2) deve essere scalabile in termini sia di utenti che di addetti ai lavori, (3) deve raggiungere un equilibrio tra costi, tempi richiesti, funzionalità e qualità accettabile.
Prerequisiti	È fortemente consigliata l'avvenuta frequenza e lo studio dei seguenti insegnamenti del I anno: Programmazione, Laboratorio di Informatica, Linguaggi di Programmazione. Dall'insegnamento di Programmazione: linguaggi imperativi, problem solving, algoritmi fondamentali. Dall'insegnamento di Laboratorio di Informatica: stili di programmazione, debugging, programmazione modulare, documentazione del codice. Dall'insegnamento di Linguaggi di Programmazione: espressioni regolari, paradigmi di programmazione
Contenuti di insegnamento (Programma)	<ul style="list-style-type: none">– Introduzione all'ingegneria del software– Processi software e sviluppo agile; Scrum– Controllo di versione; git; GitHub; GitHub Flow– Automazione dello sviluppo e del rilascio– Modellazione con UML– Analisi dei requisiti; modellare il dominio; modellare il workflow– System design; stili architetturali– OO design; la tassonomia Entity-Control-Boundary; modellare la struttura; modellare il comportamento; design pattern– Testing; analisi statica del codice; JUnit Testing Framework– Manutenzione ed evoluzione
Testi di riferimento	<ul style="list-style-type: none">– Craig Larman. Applicare UML e i pattern. Pearson. Altri libri consigliati (per approfondimenti): <ul style="list-style-type: none">– Bernd Bruegge, Dutoit Allen. Object-Oriented Software Engineering Using UML, Patterns, and Java. Pearson New Int. Ed. (Inglese)– Titus Winters, Tom Manshreck, Hyrum Wright. Software Engineering at Google. O'Reilly Media. ISBN: 9781492082798. March 2020. https://abseil.io/resources/swe-book/html/toc.html– Martina Seidl, Marion Scholz, Christian Huemer, Gerti Kappel. UML @ Classroom: An Introduction to Object-Oriented Modeling, Springer.– Martin Fowler. UML Distilled. Pearson. Gli studenti che lo desiderano possono ottenere i testi in prestito dalla Biblioteca. Può convenire verificarne la disponibilità mediante il Sistema Bibliotecario di Ateneo https://opac.uniba.it/easyweb/w8018/index.php? e contattare la biblioteca per concordare il prestito.
Note ai testi di riferimento	I libri di testo sono integrati con gli appunti presi a lezione, con le slide del docente e puntatori a risorse online disponibili sulla piattaforma e-Learning UNIBA (v. sopra 'sede virtuale').



Organizzazione della didattica			
Ore			
Totali	Didattica frontale	Laboratorio	Studio individuale
225 ore	56 ore	30 ore	139 ore
CFU/ETCS			
9 CFU	7 CFU	2 CFU	

Metodi didattici	
	Lezioni frontali supportate da slide, esercitazioni guidate dal docente in aula, laboratorio di tipo Bring-Your-Own-Device BYOD con assegnazioni di sviluppo e modellazione verificate in aula.

Risultati di apprendimento previsti	
Conoscenza e capacità di comprensione	<ul style="list-style-type: none">– Conoscere principi fondanti dell'ingegneria del software– Conoscere le diverse tipologie di applicazioni software– Conoscere le differenze tra i processi di sviluppo del software, in particolare tra modello waterfall e modello agile– Conoscere i diversi aspetti modellabili del software e le possibili finalità della modellazione– Conoscere le soluzioni di successo a problemi noti e ricorrenti nella progettazione del software
Conoscenza e capacità di comprensione applicate	<ul style="list-style-type: none">– Sviluppare competenze applicabili allo sviluppo ed evoluzione di sistemi software, aventi caratteristiche di qualità, economicità e tempestività.– Sviluppare la capacità di creare modelli, sia con prospettiva concettuale che con prospettiva software.– Acquisire familiarità con lo sviluppo di sistemi software la cui realizzazione richiede lo sforzo congiunto e prolungato di un team di lavoro.
Competenze trasversali	<p>Autonomia di giudizio</p> <ul style="list-style-type: none">– Mostrare di aver acquisito autonomia di giudizio sulle scelte relative allo sviluppo dei sistemi software. <p>Abilità comunicative</p> <ul style="list-style-type: none">– Mostrare di essere in grado di comunicare in modo appropriato le caratteristiche di prodotto e di processo nello sviluppo del software. <p>Capacità di apprendere in modo autonomo</p> <ul style="list-style-type: none">– Mostrare di aver sviluppato capacità di intraprendere in autonomia ulteriori approfondimenti su argomenti attinenti all'ingegneria del software.



Valutazione	
Modalità di verifica dell'apprendimento	<p>Prova di laboratorio e prova orale.</p> <ul style="list-style-type: none">– La prova di laboratorio consiste in esercizi di testing e modellazione UML, in modalità forward engineering o reverse engineering a partire da codice Java, utilizzando git e GitHub. Potranno essere richiesti modelli con prospettiva concettuale e modelli con prospettiva software,– La prova orale consiste nel rispondere a domande sui contenuti di insegnamento.– La prova di laboratorio e la prova orale devono essere sostenute e superate nello stesso appello. <p>Per gli studenti che frequentano attivamente è possibile svolgere un'attività di progetto come esonero dalle prove di laboratorio e orale. Il progetto consiste nel completare in gruppo, durante lo svolgimento delle lezioni, lo sviluppo di un sistema software assegnato dal docente, seguendo un processo anche questo dettato dal docente. Un progetto valutato positivamente non può essere sottoposto a ulteriore valutazione. La valutazione positiva di un progetto è valida per tutti gli appelli dell'anno accademico corrente.</p>
Criteri di valutazione	<p>Saranno valutati i risultati di apprendimento previsti così come descritti precedentemente.</p> <p>Conoscenza e capacità di comprensione:</p> <ul style="list-style-type: none">– Capacità di comprendere le domande formulate per la prova orale e rispondere in maniera pertinente e completa.– Capacità di comprendere le linee guida per lo svolgimento delle attività di laboratorio e del progetto. <p>Conoscenza e capacità di comprensione applicate:</p> <ul style="list-style-type: none">– Conoscenza esaustiva degli argomenti oggetto del programma e loro utilizzo nello svolgimento di esercizi oggetto della prova di laboratorio e nella realizzazione del progetto– Autonomia di giudizio:– Capacità di giudicare i vantaggi e gli svantaggi delle decisioni di progetto in relazione alle alternative disponibili. <p>Abilità comunicative:</p> <ul style="list-style-type: none">– Capacità di rispondere ai quesiti formulati per la prova orale in maniera corretta, completa e utilizzando appropriatamente il linguaggio tecnico.– Capacità di apprendere:– Comprensione dei contenuti del programma e capacità di utilizzare i concetti appresi nello svolgimento di esercizi di laboratorio e nella realizzazione del progetto.
Criteri di misurazione dell'apprendimento e di attribuzione del voto finale	<p>Sia la prova di laboratorio che la prova orale sono valutate in trentesimi. Il voto finale è la media aritmetica delle due prove.</p> <p>La valutazione dell'attività di progetto è in trentesimi ed è condivisa dall'intero gruppo di lavoro. La valutazione individuale è quindi corretta con punti di bonus o malus in relazione al contributo dato al gruppo e all'intera classe.</p>
Altro	Si suggerisce agli studenti di affidarsi esclusivamente alle informazioni/comunicazioni fornite sui siti ufficiali del Dipartimento di Informatica,



ovvero sui gruppi social solo se costituiti e amministrati esclusivamente dai docenti dei relativi insegnamenti:

- <https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea>
- <https://www.uniba.it/it/ricerca/dipartimenti/informatica>
- <https://elearning.uniba.it/>

I programmi degli insegnamenti sono disponibili qui:

- <https://elearning.uniba.it/>

Le informazioni che tutti gli studenti dovrebbero conoscere sono scritte nei Regolamenti didattici e manifesti degli studi disponibili nel sito:

- <https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea>

Si suggerisce agli studenti di diffidare delle informazioni e dei materiali circolanti su siti o gruppi social non ufficiali, poiché spesso sono risultati non affidabili, non corretti o incompleti. Per ogni dubbio, chiedere un incontro al docente secondo le modalità previste per il ricevimento.

Ulteriori informazioni sono disponibili nelle slide di presentazione scaricabili dalla piattaforma e-Learning UNIBA - <https://elearning.uniba.it/>