



# UNIVERSITÀ DI PISA

---

## INGEGNERIA DEL SOFTWARE

### LAURA SEMINI

Anno accademico 2020/21  
CdS INFORMATICA  
Codice 271AA  
CFU 6

|                         |           |         |     |              |
|-------------------------|-----------|---------|-----|--------------|
| Moduli                  | Settore/i | Tipo    | Ore | Docente/i    |
| INGEGNERIA DEL SOFTWARE | INF/01    | LEZIONI | 48  | LAURA SEMINI |

#### Obiettivi di apprendimento

##### *Conoscenze*

Lo studente che completerà con successo il corso:

- avrà la possibilità di utilizzare moderni strumenti e ambienti per l'analisi, la progettazione e la verifica dei sistemi software;
- sarà in grado di dimostrare una solida conoscenza di molte tecniche di modellazione ampiamente utilizzate nello sviluppo del software;
- avrà le competenze per iniziare a lavorare in un ambiente cooperativo.

##### *Modalità di verifica delle conoscenze*

Esame scritto e orale, durante COVID progetto e orale

##### *Capacità*

Lo studente sarà in grado di realizzare e documentare un progetto sw

##### *Modalità di verifica delle capacità*

Esame scritto e orale, durante COVID progetto e orale

##### *Comportamenti*

Lo studente potrà saper gestire responsabilità di conduzione di un team di progetto.

##### *Modalità di verifica dei comportamenti*

Esame orale

#### Prerequisiti (conoscenze iniziali)

Programmazione OO

#### Indicazioni metodologiche

Attività didattiche:

- Frequentando lezioni
- Partecipazione alle discussioni
- Studio individuale

Presenza: consigliato

Metodi di insegnamento:

- lezioni

Programma (contenuti dell'insegnamento)



## UNIVERSITÀ DI PISA

---

La parte iniziale fornisce motivazioni per il resto: discutiamo i problemi che si verificano nella produzione del software e nei modelli di ciclo di vita. La maggior parte è dedicata alla modellazione del prodotto software in UML, presentando i principali tipi di modelli per ciascuna attività: domini e requisiti, analisi architettoniche e dettagliate, sia dal punto di vista statico (strutturale) sia dal punto di vista dinamico (comportamentale). Infine, presentiamo le tecniche di progettazione per la verifica del software.

### Bibliografia e materiale didattico

Disponibili in buona parte online alla pagina didawiki del corso

1. Lucidi delle lezioni
2. Object Oriented and Classical Software Engineering, Stephen R. Schach, Fifth edition, McGraw Hill, Capitoli: 1-3-10
3. Object-Oriented Software Engineering, David C. Kung, Capitolo 2
4. UML@Classroom, Martina Seidl, Marion Scholz, Christian Huemer, Gerti Kappel, Springer Verlag, 2015. Disponibile per gli studenti di unipi a [questa pagina](#).
5. [Dispensa di architettura e progettazione di dettaglio](#).
6. Capitoli: 1-2-9-10-11-12-16-17-18 del libro "Software Testing and Analysis: Process, Principles, and Techniques" di Mauro Pezzè e Michal Young, nella versione free che potete richiedere seguendo le istruzioni su Teams

### Modalità d'esame

- L'esame è composto da una prova scritta ed una prova orale.
- La prova scritta consiste in:  
alcune domande o esercizi da risolvere su un caso di studio, si svolge in un'aula normale, dura 2 ore, rimane valida per la sessione d'appello in corso.
- La prova scritta è superata se:  
la totalità degli esercizi sarà stata svolta in modo sufficiente, o comunque se un eventuale esercizio non svolto sarà compensato dagli altri, svolti in modo buono o ottimo
- La prova orale consiste in:  
un colloquio tra il candidato e il docente.
- La prova orale è superata se:  
il candidato dimostra una buona comprensione dei temi del corso.

*Durante COVID: istruzioni alla pagina didawiki*

*Ultimo aggiornamento 10/09/2020 18:20*