

# Progetto del software

## Prova scritta – 20 febbraio 2026 – 2h

### PARTE 1 – RISPOSTA SINGOLA

Ogni domanda ha una sola risposta VERA.

- Una risposta esatta fa acquisire il punteggio positivo riportato a fianco della domanda
- Una risposta errata fa perdere il punteggio negativo riportato a fianco della domanda
- Una risposta lasciata in bianco viene valutata 0

1. **(3, -.5)** I diagrammi UML delle Classi:
  - a) Servono a definire i casi d'uso e gli scenari del progetto
  - b) Servono a definire la struttura di classi e interfacce del progetto
  - c) Servono a definire la struttura delle sole classi del progetto, ma non delle interfacce
  - d) Servono a definire i posti a sedere degli studenti in aula
2. **(3, -.5)** Nella progettazione top-down:
  - a) Non è mai possibile partire da librerie e framework pre-esistenti
  - b) Si parte sempre da librerie e framework pre-esistenti
  - c) Si può partire da librerie e framework pre-esistenti, a patto che siano compatibili con l'architettura scelta e non introducano un "debito di codice"
  - d) Nessuna delle precedenti
3. **(3, -.5)** Il flusso di lavoro di Git:
  - a) Prevede che esista un unico server centralizzato
  - b) Prevede che esista un unico server centralizzato, detto *origin*
  - c) Prevede che esistano multipli server di riferimento, e che uno di loro, detto *origin*, sia quello "principale"
  - d) Prevede che esistano multipli server di riferimento, e che il principale (detto *origin*) sia su GitHub o GitLab

**PARTE 2 – (POSSIBILI) RISPOSTE MULTIPLE**  
**Ogni domanda può avere da zero a quattro risposte CORRETTE.**

- **Ogni risposta esatta viene calcolata: +1**
- **Ogni risposta errata viene calcolata: -0.5**
- **Una risposta lasciata in bianco viene calcolata: 0**

4. Il principio “L - *Liskov Substitution*” del paradigma SOLID:
  - a) Prevede che si debba creare un’interfaccia per ogni cosa
  - b) Prevede che si debba incapsulare una funzionalità in una singola classe
  - c) Prevede che si debba implementare l’*Inversion of Control*
  - d) Dimostra, ad esempio, che nell’informatica un Quadrato non è un Rettangolo
5. La qualità di un progetto software:
  - a) Dipende solo dalla corretta implementazione della funzionalità richiesta, e non dal processo di sviluppo
  - b) Dipende solo dal processo di sviluppo, e non dalla corretta implementazione della funzionalità richiesta
  - c) Dipende sia dalla corretta implementazione della funzionalità richiesta che dal processo di sviluppo
  - d) Nessuna delle precedenti
6. Nel testing *in the small*:
  - a) Si lavora sui singoli moduli/micro-funzionalità del progetto
  - b) Si può lavorare sull’integrazione dei singoli moduli del progetto, ma non sugli specifici casi d’uso “a livello globale”
  - c) Non si lavora mai con sistemi asincroni
  - d) Non si lavora mai con sistemi sincroni
7. Lo stile di programmazione *fluent*:
  - a) Si basa su un’interfaccia funzionale che prevede concatenazione dei metodi (*method chaining*)
  - b) Peggiora la leggibilità del codice
  - c) Non è applicabile nei linguaggi ad oggetti
  - d) E’ più complesso da debuggare
8. Il concetto di TRL – *Technology Readiness Level*:
  - a) Non è applicabile nei progetti software di piccole dimensioni
  - b) Definisce la maturità di una tecnologia in una scala che va da 1 a 9
  - c) Definisce la maturità di una tecnologia in relazione al dominio operativo (*Operational Design Domain* - ODD)
  - d) Definisce la maturità di una tecnologia in funzione del linguaggio di programmazione che sto usando

### PARTE 3 – DOMANDE APERTE

- Una risposta esatta fa acquisire il punteggio positivo riportato a fianco della domanda
- Una risposta errata può eventualmente causare una penalità che dipende dalla gravità dell'errore
- Una risposta lasciata in bianco viene calcolata: 0
- L'eventuale sforamento del limite di righe o parole (laddove imposto), porterà a una decurtazione di un punto per ogni riga. Eventuali schematici e listati di codice non verranno presi in considerazione nel calcolo delle righe
- **SI RICORDA CHE L'UNICO FOGLIO DA CONSEGNARE E' IN CALCE AL COMPITO. QUESTO FOGLIO, PUO' SERVIRE ESCLUSIVAMENTE COME "BRUTTA COPIA". EVENTUALI RISPOSTE SCRITTE IN QUESTO FOGLIO NON VERRANNO PRESE IN CONSIDERAZIONE**

9. **(8 pt)** Si descrivano i vantaggi e gli svantaggi dei paradigmi sincrono ed asincrono, in non più di 5 righe.

Nome e Cognome \_\_\_\_\_ Matricola: \_\_\_\_\_

10. **(5 pt)** Si descriva il principio di *Single Responsibility*.

Nome e Cognome \_\_\_\_\_ Matricola: \_\_\_\_\_

## Progetto del software

### Prova scritta – 20 febbraio 2026– 2h

Indicare le risposte corrette apponendo una croce nella casella corrispondente. Per superare la prova bisogna aver raggiunto almeno 9 punti nelle domande a risposta singola/multipla, ed almeno 15 complessivamente. Questa è l'unica pagina che dovete consegnare.

	Risposte				Punti/ Penalità	
	A	B	C	D		
1					3	-0.5
2					3	-0.5
3					3	-0.5
4						
5						
6						
7						
8						

Risposta alla domanda 9 (8 pt):

Nome e Cognome \_\_\_\_\_ Matricola: \_\_\_\_\_

**Risposta alla domanda 10 (5 pt):**