

Progetto del software

Prova scritta – 14 gennaio 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 requisiti non funzionali di un software:

- a) Sono quelli relativi alla manutenibilità, alla scalabilità ed altre proprietà che non danno un valore esplicito per il cliente, e che pertanto possono venire ignorati in fase di stesura dei requisiti
- b) Sono quelli relativi alla manutenibilità, alla scalabilità ed altre proprietà che non danno un valore esplicito per il cliente, ma che vanno comunque esplicitati in fase di stesura dei requisiti
- c) Sono quelli riguardanti le funzionalità del software e che danno valore al software stesso, e pertanto possono venire ignorati in fase di stesura dei requisiti
- d) Sono quelli riguardanti le funzionalità del software e che danno valore al software stesso, e vanno pertanto esplicitati in fase di stesura dei requisiti

2. **(3, -.5)** Il principio di *design of control*

- a) Non è applicabile nella progettazione di sistemi ad oggetti
- b) Richiede di concentrarsi sugli attori del software, chiedendosi “chi fa cosa?”
- c) Prevede che non si debba conoscere la struttura architetturale del software
- d) Non è mai previsto, perché creerebbe una complicazione non necessaria nella fase di sviluppo

3. **(3, -.5)** Il principio *Single responsibility* della programmazione SOLID:

- a) Si applica solo alla programmazione funzionale
- b) Si può applicare anche alla programmazione funzionale
- c) Dichiara che una classe non può estendere le caratteristiche di una classe padre
- d) Dichiara che una classe può estendere una classe padre, a patto che possano venir sempre sostituite l'una con l'altra

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. L'architettura dotNet:
 - a) Implementa un'architettura a middleware
 - b) Non prevede l'utilizzo di librerie o di componenti esterni
 - c) Prevede l'utilizzo del solo linguaggio C#
 - d) Non prevede mai l'utilizzo del linguaggio C++
5. Il design pattern *Adapter*:
 - a) Non è applicabile ai sistemi embedded, perché non esiste il concetto di "protocolli differenti", in tali sistemi
 - b) Non è applicabile ai sistemi embedded se non c'è un sistema operativo come GNU/Linux supportato
 - c) E' applicabile anche ai sistemi embedded, ad esempio per "fare comunicare" due protocolli di basso livello differenti
 - d) E' applicabile anche ai sistemi embedded perché non esiste il concetto di "protocolli differenti", in tali sistemi
6. Un sistema sincrono:
 - a) E' sempre basato su interrupt
 - b) E' sempre basato su *callback*
 - c) Può essere basato su chiamate a funzione
 - d) Può essere basato su chiamate a sistemi cloud
7. Le tecniche di *Dependency Injection*:
 - a) Non si adattano ai progetti di piccole dimensioni
 - b) Non si adattano ai progetti di grandi dimensioni
 - c) Hanno tipicamente un costo economico elevato
 - d) Richiedono uno sforzo da parte degli utilizzatori per adottare una metodologia di lavoro strutturata e completa
8. I diagrammi UML di scenario:
 - a) Possono prevedere *pre condition* e *post condition* in relazione al cambio degli stati di un sistema
 - b) Definiscono il comportamento del sistema in relazione a come interagisce con i sistemi esterni
 - c) Sono accompagnati da disegni e rappresentazioni tabellari, chiare, univoche e complete del comportamento del sistema
 - d) Nessuna delle precedenti

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 sfioramento 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 descriva il modello di lavoro di Git, in particolare in relazione alla *working area*, alla *staging area* e ai comandi per caricare/scaricare su/da server remoto una versione di codice

Nome e Cognome _____ Matricola: _____

10. **(5 pt)** Si descriva in 5 righe il pattern MVC, oppure il pattern MVVM, a discrezione del/la candidato/a.

Nome e Cognome _____ Matricola: _____

Progetto del software

Prova scritta – 14 gennaio 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):