

FUTURA

LA SCUOLA PER L'ITALIA DI DOMANI



Ministero dell'Istruzione e del Merito
Istituto di Istruzione Secondaria Superiore
"Alessandro Greppi"

Via dei Mille 27 – 23876 Monticello B.za (LC)
www.istitutogreppi.edu.it



Programma svolto a.s. 2025-2026

Classe 4IB
Materia: Informatica
Professore Pietro Petracca
ITP: Daniele Riva

Testi adottati:

Dispense reperibili sulle piattaforme web del corso:

<https://malafrente.github.io/info-quarta> - a cura del Prof. Malafrente

https://istitutogreppi.sharepoint.com/sites/Informatica_4IB_2025_2026

Gli esempi e gli esercizi svolti durante il corso sono disponibili su:

https://github.com/InfoGreppi2020/RepoInfo4IB_2025_2026 a cura del Prof. Petracca

Fondamenti di Git

Conoscenze

- 1.1. Concetto di Version Control
- 1.2. Installazione e configurazione di Git
- 1.3. Git basics
 - 1.3.1. Git workflow
 - 1.3.2. Commit
 - 1.3.3. Git status
 - 1.3.4. Working Directory (Workspace)
 - 1.3.5. Staging Area
 - 1.3.6. Repository
 - 1.3.7. Amend, .gitignore
 - 1.3.8. Git Branching
 - 1.3.9. HEAD
 - 1.3.10. Spostarsi da un branch all'altro
 - 1.3.11. Merging
 - 1.3.11.1. Fast-Forward
 - 1.3.11.2. Conflict resolution
 - 1.3.12. Branch deletion
 - 1.3.13. Git Diff
 - 1.3.14. Git Stashing
 - 1.3.15. Git Checkout
 - 1.3.16. Detached HEAD State
 - 1.3.17. Git Restore
 - 1.3.18. Git Reset
 - 1.3.19. Git Revert
- 1.4. GitHub Basics
 - 1.4.1. Cloning
 - 1.4.2. Permissions
 - 1.4.3. Accesso con ssh e https a GitHub
 - 1.4.4. Remote Repository
 - 1.4.5. Remote e Origin
 - 1.4.6. Git Push
 - 1.4.7. Git Fetch
 - 1.4.8. Git Pull
 - 1.4.9. Merge Conflicts
 - 1.4.10. Public and Private Repositories
 - 1.4.11. Markdown files
 - 1.4.12. GitHub Pages
 - 1.4.13. GitHub Workflow
 - 1.4.13.1. Pull Request (PR)
 - 1.4.13.2. Fork
 - 1.4.13.3. Clone

Abilità

Conoscere i comandi di base per operare con Git e GitHub

Competenze

Usare i comandi di base di Git e GitHub per sviluppare un progetto software anche in team

Delegati e Lambda Expressions

Conoscenze

- 1.5. Delegates in C#
- 1.6. Invoking Delegate
- 1.7. Multicast Delegates
- 1.8. Covariance and Contravariance of Delegates
- 1.9. Lambda Expressions
- 1.10. Function Delegate and its use with lambda
- 1.11. Action Delegate and its use with lambda
- 1.12. Concetto di Closure

Abilità

Saper scrivere delegati ed espressioni lambda

Competenze

Saper scrivere delegati ed espressioni lambda per l'esecuzione di funzioni applicate in contesti di problem solving.

LINQ

Conoscenze

- 1.1. Finalità e uso del LINQ
- 1.2. Vantaggi del LINQ
- 1.3. Espressioni LINQ mediante notazione Fluent (metodi)
- 1.4. Applicazione del LINQ alle Collections
- 1.5. Principali operatori LINQ

Classification	Query Operators
Filtering	Where, OfType
Sorting	OrderBy, OrderByDescending, ThenBy, ThenByDescending, Reverse
Grouping	GroupBy, ToLookup
Join	Join
Projection	Select
Aggregation	Average, Count, Max, Min, Sum
Quantifiers	All, Any, Contains
Elements	ElementAt, ElementAtOrDefault, First, FirstOrDefault, Last, LastOrDefault, Single, SingleOrDefault
Partitioning	Skip
Concatenation	Concat
Conversion	AsEnumerable, Cast, ToArray, ToList

Abilità

Saper scrivere espressioni LINQ con l'uso dei metodi (Fluent API) applicate alle Collections.

Competenze

Applicare le espressioni LINQ e lambda per risolvere problemi di manipolazione di dati contenuti in Collections.

Persistenza dei dati: Database, ORM, EF Core

Conoscenze

- 1.1. Approccio informale ai database con SQLite
- 1.2. Concetto di database relazionale
 - 1.2.1. Utilizzo di SQLite in applicazioni
 - 1.2.2. Accesso in lettura/scrittura a SQLite, mediante interfaccia grafica con il software DB browser - <http://sqlitebrowser.org/>
- 1.3. Concetto di ORM (Objec Relational Mapper)
- 1.4. EF Core
 - 1.4.1. Concetti di base e utilizzo in applicazioni .NET Core
 - 1.4.2. CRUD operations su un database (Create, Read, Update, Delete)
 - 1.4.3. Creazione di modelli dei dati con EF Core
 - 1.4.3.1. Tipi di collegamenti tra le tabelle dei database relazionali
 - 1.4.3.2. Associazioni uno a uno, uno a molti, molti a molti
 - 1.4.3.3. Notazione Fluent per la descrizione dei vincoli delle entità e delle relazioni
 - 1.4.3.4. Tipi di dato
 - 1.4.3.5. Convertitori di valore
 - 1.4.4. Tipi di dato di SQLite (storage classes and data types)
 - 1.4.5. Migrazione di un database da riga di comando di Visual Studio
 - 1.4.5.1. Comandi Add-Migration e Update-Database

Abilità

Saper scrivere un data model utilizzando il formalismo delle classi C# e le notazioni specifiche di EF Core.

Saper effettuare la migrazione di un database a partire dal data model.

Saper connettere un'applicazione .NET Core ad un database SQLite.

Competenze

Saper scrivere applicazioni .NET Core Console che effettuano operazioni CRUD su un database SQLite.

Multithreading, TPL e Programmazione Asincrona

Conoscenze

- 1.1. Thread e threading
- 1.2. Processi e Thread a confronto
- 1.3. PCB e TCB
- 1.4. Quando è opportuno usare i thread
- 1.5. Modello dei thread del C#
- 1.6. Foreground and Background Threads
- 1.7. Passaggio di parametri a un thread
- 1.8. Ciclo di vita di un thread
- 1.9. Sospensione e interruzione di un thread
- 1.10. Concetto di threadpool
- 1.11. Il problema delle risorse condivise tra thread
- 1.12. Race condition
- 1.13. Sezione critica
- 1.14. Tecniche di sincronizzazione tra thread
 - 1.14.1. Interlocked
 - 1.14.2. Data Partitioning
 - 1.14.3. Wait Based Synchronization
 - 1.14.3.1. Monitor e lock
 - 1.14.3.2. Mutex
 - 1.14.3.3. Semaphore
 - 1.14.3.4. SemaphoreSlim
 - 1.14.4. Algoritmi e modelli notevoli associati ai thread

- 1.14.4.1. Modello produttore-consumatore
- 1.14.4.2. Algoritmo del Barbiere
- 1.15. Concetto di deadlock
- 1.16. Task Parallel Library (TPL)
- 1.17. Costrutti per la programmazione concorrente con riferimento a thread e TPL
 - 1.17.1. fork/join
 - 1.17.2. join(count)
 - 1.17.3. cobegin/coend
- 1.18. C# Task
 - 1.18.1. Programmazione asincrona basata su attività
 - 1.18.2. Attached and Detached Child Tasks
 - 1.18.3. Cancellazione di un'attività
 - 1.18.4. Cancellare un task e i suoi figli
 - 1.18.5. Gestione delle eccezioni con i task
- 1.19. Programmazione asincrona con async e await
- 1.20. Iniziare più attività contemporaneamente
- 1.21. Composizione di attività
- 1.22. Attendere le attività in modo efficiente
 - 1.22.1. WhenAll e WhenAny
 - 1.22.2. Flusso di controllo in programmi asincroni
- 1.23. Modelli di programmazione asincrona
- 1.24. Task-based asynchronous pattern (TAP)
- 1.25. I/O di file asincrono

Abilità

Riconoscere i vantaggi e i pattern di riferimento della programmazione concorrente e asincrona.

Saper utilizzare i thread e i semafori.

Saper utilizzare i Task e i metodi della libreria TPL.

Competenze

Applicare i concetti e i costrutti della programmazione parallela e asincrona per realizzare applicazioni .NET Core Console che gestiscono contemporaneamente più flussi di lavoro.

Programmazione di rete in .NET

Conoscenze

- 1.1. Concetto di URI e URL
- 1.2. Nomi di dominio e indirizzi IP
- 1.3. Network Reference Models
- 1.4. OSI e TCP/IP
- 1.5. Well known ports
- 1.6. Protocollo HTTP
 - 1.6.1. Richieste e risposte
 - 1.6.2. Request methods
 - 1.6.3. Status codes
 - 1.6.4. Message format
 - 1.6.5. Richieste e risposte HTTP in .NET
 - 1.6.5.1. Gestione del DNS
 - 1.6.5.2. Costruzione di una URI
 - 1.6.5.3. Classe HttpClient
 - 1.6.5.3.1. Scaricare del testo dalla rete
 - 1.6.5.3.2. Eseguire più richieste Web in parallelo tramite async e await
 - 1.6.5.3.3. Analizzare la risposta di una Get http mediante le proprietà della risposta
 - 1.6.5.3.4. Scaricamento e salvataggio di una pagina web
 - 1.6.5.3.5. Lettura e scrittura di file scaricati da Internet con metodi asincroni

1.6.5.3.6. Gestione di file di grandi dimensioni

1.7. Extension Methods e librerie in C#

1.7.1. Scrittura di metodi di estensione

1.7.2. Scrittura di librerie

1.7.2.1. Scrittura di una libreria per la gestione del proxy di rete

1.8. JavaScript Object Notation – JSON

1.8.1. Concetti di base

1.8.2. Nested Objects

1.8.3. Nested Arrays

1.8.4. Serializzazione e deserializzazione di oggetti JSON in .NET con System.Text.Json

Abilità

Saper utilizzare la classe HttpClient per effettuare richieste HTTP (GET, POST, PUT, DELETE) verso un host remoto.

Saper utilizzare la classe HttpClient per gestire le risposte HTTP da un host remoto.

Applicare i pattern della programmazione asincrona alle richieste HTTP.

Competenze

Saper scrivere applicazioni .NET Core Console che si connettono ad endpoints remoti ed effettuano chiamate e risposte http, utilizzando i principali verbi GET, POST, PUT, DELETE.

REST API

1.1. REST (REpresentational State Transfer) Api

1.2. Principi guida dell'architettura REST

1.3. Consumare REST API mediante l'oggetto HttpClient

1.4. CRUD operations con REST API

1.5. Utilizzo di Postman

1.5.1. Testing di Api con Postman

1.5.2. Collections di Postman

1.5.3. Autorizzazioni in Postman

1.6. Esempi di servizi REST

1.6.1. Weather API (<https://open-meteo.com/>)

1.6.1.1. Recupero dei dati relativi alle condizioni meteo correnti

1.6.1.2. Previsioni meteo a 7 giorni e previsioni dettagliate a 48 ore

Abilità

Saper implementare “CRUD operations” verso un host HTTP remoto conforme alle specifiche REST.

Saper consumare REST API di terze parti.

Competenze

Saper scrivere un'applicazione .NET Core Console in grado di consumare servizi REST di terze parti.

.NET MAUI

Conoscenze

1.1. .NET MAUI Getting Started (Conoscenze di base sull'architettura, ambiente di sviluppo, applicazione demo, creazione di una interfaccia grafica in XAML per MAUI)

<https://learn.microsoft.com/en-us/training/modules/build-mobile-and-desktop-apps/>

<https://dotnet.microsoft.com/en-us/learn/maui/first-app-tutorial/intro>

<https://learn.microsoft.com/en-us/training/modules/create-user-interface-xaml/>

1.2. Pagine e Navigazione

1.2.1. Content Page

<https://learn.microsoft.com/en-us/dotnet/maui/user-interface/pages/contentpage>

1.2.2. Navigazione gerarchica - NavigationPage

<https://learn.microsoft.com/en-us/dotnet/maui/user-interface/pages/navigationpage>

- 1.2.3. Shell
<https://learn.microsoft.com/en-us/dotnet/maui/fundamentals/shell>
- 1.3. Tipi di Layout di Pagina
 - 1.3.1. Grid
<https://learn.microsoft.com/en-us/dotnet/maui/user-interface/layouts/grid>
 - 1.3.2. AbsoluteLayout
<https://learn.microsoft.com/en-us/dotnet/maui/user-interface/layouts/absolutelayout>
 - 1.3.3. FlexLayout
<https://learn.microsoft.com/en-us/dotnet/maui/user-interface/layouts/flexlayout>
 - 1.3.4. StackLayout, HorizontalStackLayout, VerticalStackLayout
<https://learn.microsoft.com/en-us/dotnet/maui/user-interface/layouts/stacklayout>
<https://learn.microsoft.com/en-us/dotnet/maui/user-interface/layouts/horizontalstacklayout>
<https://learn.microsoft.com/en-us/dotnet/maui/user-interface/layouts/verticalstacklayout>
- 1.4. Persistenza in MAUI
 - 1.4.1. Salvare in maniera persistente i dati localmente sul dispositivo.
<https://learn.microsoft.com/en-us/training/modules/store-local-data/2-compare-storage-options>
 - 1.4.2. Preferences
<https://learn.microsoft.com/en-us/dotnet/maui/platform-integration/storage/preferences>
<https://learn.microsoft.com/en-us/dotnet/api/microsoft.maui.storage.preferences>
 - 1.4.3. Local File System (File System Helpers)
<https://learn.microsoft.com/en-us/dotnet/maui/platform-integration/storage/file-system-helpers>
 - 1.4.4. Cache directory
<https://learn.microsoft.com/en-us/dotnet/maui/platform-integration/storage/file-system-helpers#cache-directory>
 - 1.4.5. App data directory
<https://learn.microsoft.com/en-us/dotnet/maui/platform-integration/storage/file-system-helpers#app-data-directory>
 - 1.4.6. Bundled files
<https://learn.microsoft.com/en-us/dotnet/maui/platform-integration/storage/file-system-helpers#bundled-files>
- 1.5. Data Binding
 - 1.5.1. Data Binding and MVVM
<https://learn.microsoft.com/en-us/dotnet/maui/xaml/fundamentals/mvvm>
 - 1.5.2. MVVM Community Toolkit
<https://learn.microsoft.com/en-us/dotnet/communitytoolkit/mvvm/>
<https://learn.microsoft.com/en-us/dotnet/communitytoolkit/mvvm/generators/overview>
<https://learn.microsoft.com/en-us/dotnet/communitytoolkit/mvvm/generators/relaycommand>
<https://learn.microsoft.com/en-us/dotnet/communitytoolkit/mvvm/generators/inotifypropertychanged>
<https://learn.microsoft.com/en-us/dotnet/communitytoolkit/mvvm/generators/observableproperty>
- 1.6. Local database in .NET MAUI (SQLite con il pacchetto sqlite-net-pcl)
 - 1.6.1. Aspetti teorici:
<https://learn.microsoft.com/en-us/dotnet/maui/data-cloud/database-sqlite>
 - 1.6.2. Esempi:
<https://learn.microsoft.com/en-us/samples/dotnet/maui-samples/database-sqlite/>
 - 1.6.3. Esercitazione guidata:
<https://learn.microsoft.com/en-us/training/modules/store-local-data/>
 - 1.6.4. <https://github.com/MicrosoftDocs/mslearn-dotnetmaui-store-local-data>

Abilità

Saper descrivere l'architettura di un'applicazione mobile sviluppata in .NET MAUI.

Saper scrivere il codice per implementare ciascuno dei componenti e dei meccanismi di funzionamento descritti nella sezione conoscenze.

Competenze

Saper realizzare applicazioni Android e Windows complete con i componenti di .NET MAUI.

Sviluppo di App assistito da AI

- 1.1. Vibe Coding vs Sviluppo Spec-Driven
- 1.2. Il modello operativo: Man-in-the-Loop
- 1.3. Struttura documentale del progetto (Docs-as-Code)
 - 1.3.1. File AGENTS.md
 - 1.3.2. File spec.md
 - 1.3.3. File plan.md
 - 1.3.4. File architecture.md
 - 1.3.5. File prompt-log.md
 - 1.3.6. File test-matrix.md
 - 1.3.7. Files iterations/it-template.md
 - 1.3.8. File deployment.md
- 1.4. Strumenti per lo sviluppo assistito da AI
 - 1.4.1. Claude Code
 - 1.4.2. OpenCode
 - 1.4.3. GitHub Copilot
- 1.5. Fasi di sviluppo del progetto MAUI
 - 1.5.1. Analisi
 - 1.5.2. Design e mockup
 - 1.5.3. Pianificazione iterativa
 - 1.5.4. Uso di OpenCode durante lo sviluppo
 - 1.5.5. Testing
 - 1.5.6. Sicurezza e uso responsabile dell'AI
- 1.6. Skills per agenti AI di coding
 - 1.6.1. Struttura completa di una Skill
 - 1.6.2. L'ecosistema skills.sh: il registry delle skills
 - 1.6.3. Le skills più importanti nell'ecosistema
 - 1.6.3.1. find-skills: la skill per trovare altre skills
 - 1.6.3.2. skill-creator: la meta-skill per creare nuove skills
 - 1.6.4. Come le skills vengono utilizzate dall'agente AI
 - 1.6.5. Esempi pratici di creazione di skill
 - 1.6.5.1. Una skill per lo sviluppo di applicazioni .NET MAUI secondo le best practices - skill maui-expert
 - 1.6.5.2. Una skill dal proprio workflow con skill-creator – skill man-in-the-loop-workflow
 - 1.6.5.3. Una skill per la generazione di un documento di specifica dettagliato a partire dall'elenco dei requisiti funzionali e non funzionali di una app – skill maui-prd
 - 1.6.5.4. Una skill per trasformare un documento di specifica in un piano iterativo di lavoro – skill prd-to-plan
 - 1.6.5.5. Una skill per il testing di un'app – skill maui-automatic-testing
- 1.7. Progettazione UI/UX con il supporto dell'AI
 - 1.7.1. Approccio basato su prompt per la UI/UX
 - 1.7.2. Approccio basato su Skills per la UI/UX
 - 1.7.3. Google Stitch per la progettazione UI
 - 1.7.3.1. DESIGN.md: il design system portabile di Stitch

Abilità

Saper utilizzare agenti AI per lo sviluppo di codice (programmazione automatica)

Competenze

Saper sviluppare un'applicazione .NET Core Console e MAUI mediante agenti AI, skills e workflow consolidati.

Attività di laboratorio

Per ogni unità didattica sono stati sviluppati numerosi esempi in laboratorio, contestualmente alle lezioni teoriche.

Monticello, li 04/06/2026

Gli Alunni

Prof. Pietro Petracca

ITP Daniele Riva

Per la valutazione delle prove scritte, pratiche e orali sono state utilizzate le griglie di valutazione riportate di seguito

GRIGLIA DI VALUTAZIONE PER LE PROVE SCRITTE E PRATICHE DI INFORMATICA

Alunno/a: _____ Classe: _____ Data: _____

Argomento della prova: _____

1. PRINCIPI DI VALUTAZIONE

La presente verifica è suddivisa in **indicatori** (esercizi, metodi o funzionalità richieste). Ad ogni indicatore è assegnato un **Peso Massimo** (punteggio massimo ottenibile).

Il punteggio effettivo assegnato a ciascun indicatore viene calcolato applicando una **percentuale di raggiungimento** (da 0% a 100%) al peso massimo. Questa percentuale viene stabilita dal docente valutando tre aspetti fondamentali del codice prodotto:

- **Correttezza (Funzionalità):** Il codice compila senza errori? Risolve il problema richiesto generando l'output atteso? Vengono gestiti correttamente i casi limite?
- **Stile e Qualità (Leggibilità):** Il codice è scritto secondo le “best practices” di settore? Le variabili e i metodi hanno nomi significativi (naming convention)? Le strutture dati e di controllo sono usate in modo appropriato?
- **Originalità e Problem Solving:** La soluzione proposta è efficiente? Lo studente ha dimostrato autonomia nel trovare soluzioni eleganti o ha ottimizzato il codice oltre la richiesta base?

Livelli di Raggiungimento (Fattore Percentuale)

Assegnata	Livello	Descrizione dei requisiti soddisfatti
90% - 100%	Eccellente	Funzionalità completa e corretta. Ottimo stile, codice pulito, leggibile e ben ottimizzato. Soluzione autonoma e brillante.
70% - 85%	Buono / Discreto	Il codice è funzionante e risolve il problema. Sono presenti piccole imprecisioni nello stile, o la logica, seppur corretta, poteva essere ottimizzata.
50% - 65%	Sufficiente	Il problema è parzialmente risolto. Il codice compila ma presenta errori logici non bloccanti, oppure lo stile è molto disordinato e poco leggibile.
20% - 45%	Insufficiente	L'esercizio è stato tentato ma presenta gravi errori di logica, errori di compilazione o copre solo una minima parte dei requisiti.
0% - 15%	Gravemente Insuff.	Indicatore non svolto, completamente errato o non valutabile.

2. ESEMPIO DI GRIGLIA DI CORREZIONE SPECIFICA DELLA PROVA

(Nota: La tabella seguente è un modello adattabile. La griglia effettiva viene stabilita in base all'argomento e alla difficoltà della prova. Si aggiungono o si rimuovono righe in base al numero di richieste della specifica traccia. La somma dei pesi massimi è pari a 10, oppure il totale andrà riproporzionato in decimi).

#	Indicatore / Parte della Prova	Peso Max	% Raggiunta	Punteggio (Peso x %)	Note / Feedback Docente
1	<i>Esercizio 1: Struttura Classi di Base</i>	1.0	____ %	_____	
2	<i>Esercizio 1: Implementazione Metodo A</i>	0.5	____ %	_____	
3	<i>Esercizio 1: Implementazione Metodo B</i>	1.0	____ %	_____	
4	<i>Esercizio 1: Menu e gestione I/O</i>	1.5	____ %	_____	
5	<i>Esercizio 2: Lettura File e Parsing</i>	1.0	____ %	_____	
6	<i>Esercizio 2: Logica di conteggio</i>	1.0	____ %	_____	
...	<i>[Aggiungere altri indicatori...]</i>	...	____ %	_____	
TOT	PUNTEGGIO TOTALE CONSEGUITO	10.0		_____ / 10	

3. CALCOLO DEL VOTO FINALE

- **Punteggio Grezzo:** _____
- **Fattore Compensativo / Disp. BES (se applicabile):** _____
- **VOTO FINALE IN DECIMI:** _____

BN: il voto minimo assegnato non potrà essere inferiore a 2.

Commento Generale del Docente:

(Spazio per un eventuale feedback discorsivo sull'andamento generale della prova, punti di forza e aree di miglioramento dello studente).

GRIGLIA DI VALUTAZIONE - PROVA ORALE DI INFORMATICA

Alunno/a: _____ Classe: _____ Data: _____

Argomento/i del colloquio: _____

1. PARAMETRI DI OSSERVAZIONE

Durante l'intero colloquio, l'esposizione dell'alunno/alunna viene valutata in modo globale tenendo conto dei seguenti tre pilastri fondamentali:

1. **Correttezza e Profondità:** Esattezza delle informazioni tecniche, livello di dettaglio e appropriatezza del lessico informatico utilizzato.
2. **Autonomia e Originalità:** Capacità di sostenere un discorso fluido senza continui solleciti, abilità nel fare collegamenti interdisciplinari o di portare esempi autonomi.
3. **Problem Solving (Ragionamento Logico):** Capacità di analizzare uno scenario pratico o del codice alla lavagna, formulare un'ipotesi risolutiva e motivare le scelte.

2. RUBRICA DI VALUTAZIONE DEL COLLOQUIO

(Barrare la riga che descrive in modo più aderente la prestazione complessiva dell'alunno/a)

Voto /10	Livello	Descrittore Globale
10	Eccellente	L'alunno/a dimostra una padronanza completa, critica e approfondita degli argomenti. L'esposizione è estremamente fluida, del tutto autonoma e utilizza un lessico tecnico rigoroso e ineccepibile. Risolve i problemi proposti in modo brillante, originale e altamente ottimizzato, dimostrando spiccate capacità logiche.
9	Distinto	Ottima padronanza degli argomenti con un buon livello di approfondimento. Esposizione fluida e autonoma con lessico tecnico preciso. Risolve i problemi in modo logico e corretto, con buona autonomia e proponendo soluzioni valide ed efficaci.
8	Buono	La conoscenza degli argomenti è sicura e chiara. L'esposizione è scorrevole e il lessico appropriato. Affronta i problemi pratici in modo logico, arrivando alla soluzione corretta in autonomia o con minimi e ininfluenti suggerimenti.
7	Discreto	Conoscenza degli argomenti adeguata, pur priva di particolari approfondimenti. L'esposizione è comprensibile e sufficientemente autonoma, con un lessico tecnico di base corretto. Risolve i problemi standard ma necessita di qualche spunto dal docente per superare gli ostacoli più complessi.
6	Sufficiente	La conoscenza degli argomenti è essenziale e limitata ai concetti base. L'esposizione, a tratti frammentaria o mnemonica, raggiunge l'obiettivo minimo. Il lessico tecnico è semplice o parzialmente insicuro. Il problem solving richiede una guida costante da parte del docente per arrivare alla soluzione.
5	Mediocre	Conoscenze parziali, lacunose e non sempre chiare. L'esposizione è incerta e fatica a fare collegamenti. Il linguaggio tecnico è spesso inappropriato. Mostra evidenti difficoltà nel ragionamento logico, riuscendo a impostare una soluzione solo se fortemente guidato, senza però completarla del tutto.
4	Insuff.	Conoscenze molto superficiali e frammentarie, con gravi errori concettuali. L'esposizione è stentata e il linguaggio inadeguato. Non è in grado di avviare un ragionamento logico per risolvere i problemi proposti, nemmeno con il forte supporto del docente.
2 - 4	Gravemente insuff.	Preparazione nulla o quasi. Capacità di ragionamento logico assenti.

Nota: il criterio di assegnazione del voto nella fascia 2-4 e per l'attribuzione del mezzo voto (ad esempio 6.5, 7.5, etc.) nelle altre fasce è il grado di raggiungimento dei descrittori indicati per il livello corrispondente.

3. ESITO DELLA PROVA

- **VOTO ASSEGNATO:** _____ / 10
 - **Applicazione Strumenti Compensativi (BES/DSA):** [] Sì [] No
- Specificare:

Note / Feedback Docente:

(Eventuali osservazioni su punti di forza e aree di miglioramento da restituire all'alunno/a)