

# 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](http://www.istitutogreppi.edu.it)



## Programma svolto a.s. 2025-2026

Classe 3IB

Materia: Informatica

Professore Pietro Petracca

ITP: Stefano Riva

Testi adottati:

[Fundamentals of computer programming with C#](#) di Svetlin Nakov & Co.

Dispense integrative, reperibili sulla piattaforma

[https://istitutogreppi.sharepoint.com/sites/Informatica\\_3IB\\_2025\\_2026](https://istitutogreppi.sharepoint.com/sites/Informatica_3IB_2025_2026) tramite le proprie credenziali.

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

[https://github.com/InfoGreppi2020/RepoInfo3IB\\_2025\\_2026](https://github.com/InfoGreppi2020/RepoInfo3IB_2025_2026)

# Programma

## ***Programmazione Strutturata***

### **Conoscenze**

Computazione, problem solving, algorithm design

Teorema di Böhm-Jacopini

Architettura del calcolatore (cenni)

Tipologie di linguaggi

Machine Languages

Assembly Languages and Assemblers

High-Level Languages, Compilers and Interpreters

The .NET Framework

CIL and JIT, Assemblies, Managed Code, Linking, Garbage Collection

Stages in Software Development

Documentation

Software Development

Primitive types and variables

Concetto di variabile

I tipi predefiniti del C#

Valori di default

Interi, float, double, decimal

Numero di cifre significative

Errori di rappresentazione

Booleano - boolean.

Carattere – char

Stringa – string

Tipi valore e tipi riferimento, Valori letterali, Var keyword

Operatori ed espressioni

Categorie di operatori

Associatività degli operatori

Precedenza degli operatori

Operatori aritmetici, logici, bit a bit, di uguaglianza e di confronto

Type conversion and casting

Espressioni

Console Input e Output

Composite Formatting – string.Format(), Console.WriteLine(),

String Interpolation

String Concatenation

## Conditional Statements

### Costrutti fondamentali

If, if .. else

Switch

Constant pattern

Type pattern

Operatore condizionale ?:

## Loops

Do while

While

For

break e continue

Tipo enumerativo – enum

Generazione di numeri casuali

## Array, Struct e Date

Definizione di array

Array multidimensionali

Parallel arrays

foreach Loops

Struct

Array di record (con struct)

Array paralleli vs. array di record

Date

DateTime - struct

TimeSpan – struct - un intervallo di tempo

Parsing di DateTime

Esempi di DateTime.Parse()

Esempi con TryParse

## Metodi

Generalità sui metodi

Passaggio dei parametri

Passaggio di parametri di tipi di valore

Passaggio di tipi valore per valore

Passaggio di tipi valore per riferimento

Passaggio di parametri di tipi di riferimento

Passaggio di tipi riferimento per valore

Passaggio di tipi riferimento per riferimento

Scambio di due stringhe

Passaggio di array mediante ref e out

Differenza tra il passaggio a un metodo di una struct e di un riferimento a un'istanza di classe

Complessità computazionale degli algoritmi.

Notazioni  $O(n)$  e confronto tra funzioni asintotiche

Problema della ricerca del minimo (massimo) di un array

Problema dell'ordinamento di un array

Algoritmi di ordinamento iterativi

Selection Sort

Insertion Sort

Bubble Sort

Ricorsione

Caratteristiche comuni di un algoritmo ricorsivo

Confronto tra algoritmi di ordinamento ricorsivi e algoritmi iterativi

Scrittura di metodi ricorsivi

Algoritmi di ordinamento ricorsivi

Merge Sort (cenni)

Quick Sort (cenni)

Ricerca di una chiave in un array ordinato (ricerca dicotomica ricorsiva)

## **Abilità**

Saper codificare un algoritmo in linguaggio C#.

Saper strutturare un algoritmo utilizzando le strutture di base del linguaggio C#

Saper scrivere metodi per la suddivisione dei compiti di un algoritmo

## **Competenze**

Saper risolvere problemi attraverso l'uso del calcolatore, usando il linguaggio C# per codificare algoritmi anche complessi.

## ***Programmazione Object Oriented***

### **Conoscenze**

Creating and using Objects

Defining Classes

Modificatori di accesso

Defaults

Namespaces vs Assemblies

Capitalization conventions

Object Oriented Programming (OOP)

Implementazione esplicita delle interfacce

La classe Object: dove tutto ha inizio

Metodi di Object

is operator in C#

as operator in C#

Metodo GetType

Confronto tra oggetti: Interfaccia IComparable

Confronto tra oggetti: Interfaccia IComparable<T>: Metodo CompareTo e overload degli operatori >, <, >=, <=

Confronti aggiuntivi: Classe astratta Comparer<T>

Uguaglianza di oggetti: Interfaccia IEquatable<T>, override di Equals() e GetHashCode() di System.Object, override degli operatori == e !=

Uguaglianze aggiuntive: interfaccia IEqualityComparer<T>

Clonazione di oggetti

MemberwiseClone()

Clonazione superficiale (shallow copy)

Clonazione profonda (deep copy)

## NET API

<https://docs.microsoft.com/it-it/dotnet/api/>

<https://docs.microsoft.com/it-it/dotnet/api/system.collections.generic>

<https://docs.microsoft.com/it-it/dotnet/api/system>

Esempi notevoli di classi e interfacce utili nelle applicazioni

List<T> Class

HashSet<T> Class

Dictionary<TKey,Tvalue> Class

Stack<T> Class

Queue<T> Class

Gestione delle stringhe in C#

Exception Handling

Exception Hierarchy

Hierarchy of Runtime Exceptions

Streams and I/O

File and Stream I/O

Files and Directories

Streams

Stream architecture

Stream adapters

Readers and Writers

Dive into files

Ottenere informazioni su un file

FileInfo class

DirectoryInfo class

Percorsi assoluti e relativi

Approccio semplice con i file di testo

Scrittura su un file di testo - StreamWriter(fileName)

Lettura da un file di testo - StreamReader(fileName)

File delimitati - CSV file

Creazione di un CSV

Lettura da un CSV

Classi di riferimento

File class

Directory class

FileStream class

Path class

Esempi di operazioni di I/O comuni

Copiare una directory e tutto il suo contenuto in un altro percorso

Cancellare una directory e tutto il suo contenuto oppure cancellare un file

Eliminare file

Rinominare una directory o un file oppure spostare una directory o file in un altro posto

Gestione dei caratteri jolly (wildcards)

Percorsi predefiniti in Windows

Applicazioni con interfaccia grafica (GUI) con .NET MAUI

Architettura di .NET MAUI

Il linguaggio XAML

Scrittura di interfacce grafiche mediante tag XAML

Widget grafici: Entry, Label, Button, Editor, Image

Layout grafici: StackLayout, GridLayout

Posizionamento di un widget all'interno di un Layout

Concetti di Margin, Padding

Modello ad eventi

Gestione degli eventi (Click)

Tutorial studiati in classe:

<https://learn.microsoft.com/en-us/training/paths/build-apps-with-dotnet-maui/>

## Abilità

Saper scrivere classi in C# con campi, properties, costruttori e metodi

Saper implementare interfacce e classi astratte

Saper applicare i principi dell'ereditarietà e del polimorfismo per creare gerarchie di classi

Saper scrivere programmi che si interfacciano con file e stream.

Saper scrivere applicazioni con interfaccia grafica con .NET MAUI

### **Competenze**

Saper scrivere programmi con tecnica Object Oriented per risolvere problemi anche complessi.

Saper modellare sistemi anche complessi con la tecnica dell'Object Oriented Programming

### ***Attività di laboratorio***

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

Monticello, 04/06/2026

Gli Alunni

---

Prof. Pietro Petracca

---

Prof. ITP Stefano Riva

---

Per le valutazioni 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%	<b>Eccellente</b>	Funzionalità completa e corretta. Ottimo stile, codice pulito, leggibile e ben ottimizzato. Soluzione autonoma e brillante.
70% - 85%	<b>Buono / Discreto</b>	Il codice è funzionante e risolve il problema. Sono presenti piccole imprecisioni nello stile, o la logica, seppur corretta, poteva essere ottimizzata.
50% - 65%	<b>Sufficiente</b>	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%	<b>Insufficiente</b>	L'esercizio è stato tentato ma presenta gravi errori di logica, errori di compilazione o copre solo una minima parte dei requisiti.
0% - 15%	<b>Gravemente Insuff.</b>	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>	...	____ %	_____	
<b>TOT</b>	<b>PUNTEGGIO TOTALE CONSEGUITO</b>	<b>10.0</b>		_____ / 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	<b>Eccellente</b>	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	<b>Distinto</b>	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	<b>Buono</b>	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	<b>Discreto</b>	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	<b>Sufficiente</b>	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	<b>Mediocre</b>	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	<b>Insuff.</b>	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	<b>Gravemente insuff.</b>	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)*