



**ISTITUTO TECNICO INDUSTRIALE "P. HENSEMBERGER"**  
ISTITUTO TECNICO : Elettrotecnica, Informatica, Telecomunicazioni,  
Meccanica e Meccatronica, Biotecnologie Sanitarie

LICEO SCIENTIFICO : Scienze Applicate

✉ e-mail [MBTF410002@istruzione.it](mailto:MBTF410002@istruzione.it) e-mail PEC [MBTF410002@pec.istruzione.it](mailto:MBTF410002@pec.istruzione.it)

Sito internet: [www.hensemberger.edu.it](http://www.hensemberger.edu.it)

## **OBIETTIVI ESSENZIALI DI DISCIPLINA**

### **Dipartimento di Informatica e Telecomunicazioni**

### **Informatica (Liceo)**

#### **PRIMO BIENNIO**

**AREA TEMATICA: Architettura dell'elaboratore e codifica dell'informazione**

#### **COMPETENZE**

Comprendere i concetti fondamentali dell'ICT (Information and Communications Technology).  
Conoscere la codifica di dati quali numeri naturali e caratteri.  
Riconoscere le componenti di un PC e le rispettive funzioni.

#### **ABILITA'**

Saper svolgere operazioni aritmetiche con numeri binari e conversioni tra diverse basi.  
Riconoscere le caratteristiche funzionali di un computer (calcolo, elaborazione, comunicazione).

#### **CONOSCENZE**

Cos'è l'informatica.  
Cosa vuol dire ICT.  
Campi di applicazione dell'informatica  
Sistemi di numerazione decimale, binario, esadecimale e rispettive conversioni.  
Informazioni, dati e loro codifica.  
Architettura e componenti di un computer.  
modello di Von Neumann.

#### **Livello base:**

Conosce la struttura di base e la funzione di ogni componente di un PC.  
Conosce il modello logico di un sistema di elaborazione.  
Sa trasformare un numero da binario a decimale e viceversa, da binario ad esadecimale e viceversa.  
Si esprime in modo chiaro.

**AREA TEMATICA: Le basi della programmazione (linguaggio di progetto)**

#### **COMPETENZE**

Individuare le strategie appropriate per la soluzione di problemi.

#### **ABILITA'**

Risolvere problemi in forma algoritmica utilizzando i diagrammi di flusso (o un linguaggio di progetto).  
Saper implementare gli algoritmi precedentemente progettati.

#### **CONOSCENZE**

Algebra booleana.  
Problemi e classi di problemi.  
Definizione di algoritmo e proprietà caratteristiche.  
Rappresentazione degli algoritmi (diagrammi di flusso).  
Schemi fondamentali: sequenza, selezione, iterazione.  
Tabella di traccia dell'esecuzione di un algoritmo.  
Struttura fondamentale di un programma.  
Variabili e costanti.  
Espressioni e condizioni.  
Operazioni standard di input e output.  
Controllo del flusso di esecuzione.  
Funzioni della libreria matematica.



Ambiente di sviluppo integrato (Dev-C++, Microsoft Visual Studio o equivalenti).  
Fasi di sviluppo di un programma.

**Livello base:**

Individua correttamente i dati in ingresso e i dati in uscita.  
Comprende le richieste del testo dopo attenta lettura ed analisi.  
Analizza e progetta semplici algoritmi in situazioni note.  
Utilizza gli strumenti proposti per la rappresentazione degli algoritmi.  
Imposta in modo corretto la tabella di traccia.  
Conosce il linguaggio tecnico appropriato per l'esposizione dei concetti.  
Saper interpretare codice scritto nel linguaggio di programmazione.

**AREA TEMATICA: Le basi della programmazione (linguaggio di programmazione)**

**COMPETENZE**

Individuare le strategie appropriate per la soluzione di problemi.

**ABILITA'**

Utilizzare un linguaggio di programmazione per implementare algoritmi (programmazione C/C++).  
Saper implementare gli algoritmi usando un linguaggio di programmazione (C/C++).

**CONOSCENZE**

Schemi fondamentali: sequenza, selezione, iterazione.  
Struttura fondamentale di un programma.  
Tipi di dato.  
Variabili e costanti.  
Espressioni e condizioni.  
Operazioni standard di input e output.  
Selezione (if, if annidati, if in cascata, switch).  
Iterazione (ciclo while, ciclo do-while, ciclo for, break, continue).  
Funzioni e parametri.  
Ambiente di sviluppo integrato (Dev-C++, Microsoft Visual Studio o equivalente).

**Livello base:**

Sa utilizzare il linguaggio progetto in modo corretto, individuando i dati in ingresso ed i dati in uscita.  
Comprende le richieste del testo dopo attenta lettura ed analisi.  
Analizza, progetta e codifica semplici algoritmi in situazioni note.  
Utilizza gli strumenti proposti per la rappresentazione degli algoritmi.  
Imposta in modo corretto la tabella di traccia.  
Sa interpretare codice scritto nel linguaggio di programmazione.  
Conosce il linguaggio tecnico appropriato per l'esposizione dei concetti.

**AREA TEMATICA: Il foglio elettronico**

**COMPETENZE**

Analizzare dati e interpretarli sviluppando deduzioni e ragionamenti sugli stessi anche con l'ausilio di rappresentazioni grafiche, usando consapevolmente gli strumenti di calcolo e le potenzialità offerte da applicazioni specifiche di tipo informatico.  
Saper utilizzare gli strumenti informatici per il proprio lavoro domestico, trasversalmente alle discipline.

**ABILITA'**

Utilizzare applicazioni elementari di calcolo.  
Raccogliere, organizzare e rappresentare informazioni.

**CONOSCENZE**

Foglio di calcolo (Excel o equivalente).  
Aprire, chiudere, salvare un foglio elettronico.  
Impostare le opzioni di base del programma.  
Operazioni e celle.



Formule e funzioni.  
Formattazione.  
Stampa.  
Grafici.

**Livello base:**

Sa creare e salvare e modificare un foglio.  
Effettua calcoli con formule semplici con riferimento alle celle.  
Ordina i dati secondo diversi criteri.  
Sa inserire, selezionare, copiare e cancellare formule e testo.  
Sa formattare le tabelle di dati e preparare i dati per la stampa.

**AREA TEMATICA: Sistemi operativi e software di base**

**COMPETENZE**

Individuare le funzioni e le competenze del software di base.  
Riconoscere i diversi tipi di linguaggi di programmazione e le rispettive caratteristiche.

**ABILITA'**

Saper usare programmi del software di base in modo appropriato.  
Classificare i linguaggi di programmazione.

**CONOSCENZE**

Definizione e funzioni del sistema operativo.  
Le funzioni del sistema operativo.  
La struttura del sistema operativo.  
File system, file e directory.  
La gestione della memoria.  
Programma e processo.  
Linguaggi di programmazione.  
I programmi traduttori: i compilatori.  
I programmi traduttori: gli interpreti.  
Le fasi di linking e loading.

**LIVELLI**

**Livello base:**

Conosce le funzione del S.O.  
Sa definire il concetto di programma e processo.  
Conosce la funzione dei principali moduli del S.O.  
Distingue i diversi tipi di linguaggi di programmazione.

**SECONDO BIENNIO**

**AREA TEMATICA: Elementi di programmazione**

**COMPETENZE**

Utilizzare i costrutti base della programmazione individuando le strategie appropriate per la soluzione di problemi.  
Utilizzare e definire appositi sottoprogrammi per la soluzione di problemi.  
Utilizzare strutture dati aggregate per la soluzione di problemi.  
Utilizzare strutture dati personalizzate.

**ABILITA'**

Impostare e risolvere problemi, anche relativi a dati aggregati, in C++.  
Sfruttare la suddivisione in sottoproblemi per la risoluzione di problemi reali.  
Impostare e implementare programmi che fanno uso di strutture dati appositamente definite.

**CONOSCENZE**



Vettori e matrici.  
Vettori e accesso in lettura e scrittura.  
Vettori sovradimensionati e gestione tramite indici.  
Vettori paralleli.  
Matrici.  
Il tipo string. Stringhe come vettori di caratteri.  
Funzioni e parametri.  
Passaggio dei parametri per valore. Cenni sul passaggio dei parametri per riferimento.  
Vettori come parametri di una funzione.  
Strutture dati definite dal programmatore (struct).  
Uso di struct per la definizione di dati strutturati con eventuali annidamenti.  
Definizione e uso di vettori di strutture.  
Cenni su compilatori e interpreti.

**Livello base:**

Comprende le richieste del testo.  
Analizza e progetta semplici algoritmi in situazioni note.  
Conosce e usa i costrutti del linguaggio C++ per codificare algoritmi in situazioni note.  
Individua correttamente i dati in ingresso/uscita.  
Suddivide semplici algoritmi in blocchi funzionali.  
Utilizza i dati aggregati in situazioni note.  
Utilizza semplici strutture appositamente definite.  
Implementa gli algoritmi relativi alle operazioni fondamentali su vettori di struct: lettura, scrittura, visita e ricerca.

**AREA TEMATICA: I file**

**COMPETENZE**

Conoscere le funzioni principali per la gestione di strutture di memorizzazione dati su memoria di massa locali.  
Risolvere problemi facendo uso di opportune strutture di memorizzazione dati su memoria di massa locali.  
Accedere e utilizzare la memoria di massa per la persistenza dei dati.

**ABILITA'**

Risolvere problemi reali usando strutture dati memorizzate su memoria di massa locale.  
Implementare la persistenza dei dati per un programma.

**CONOSCENZE**

Concetto di file.  
File in C++.  
Apertura e chiusura di un file.  
Lettura da file.  
Scrittura su file.  
File di testo.

**Livello base:**

Codifica ed implementa soluzioni a semplici problemi che fanno uso di file.  
Codifica soluzioni a problemi di semplici accessi di dati su file sia in lettura che in scrittura.

**AREA TEMATICA: Hypertext Markup Language (HTML)**

**COMPETENZA**

Strutturare informazioni in forma ipertestuale.

**ABILITA'**

Progettare, realizzare e gestire pagine web statiche.

**CONOSCENZE**



Ipertesti e multimedialità  
I tag fondamentali  
I collegamenti ipertestuali  
Le immagini  
Le tabelle

**Livello base:**

Conosce i principali tag HTML.  
Codifica pagine HTML che fa uso di tag per la formattazione del testo.  
Codifica pagine HTML contenenti collegamenti ipertestuali.

**AREA TEMATICA: La modellazione concettuale Entità-Relazione**

**COMPETENZE**

Rappresentare la struttura di un problema reale utilizzando il modello E/R.  
Conoscere i principali strumenti teorici per la modellazione e la gestione di grandi quantità di dati.

**ABILITA'**

Analizzare un problema reale e rappresentarne la struttura facendo uso di diagrammi E/R.

**CONOSCENZE**

Introduzione agli archivi e alle basi di dati.  
Entità.  
Attributi.  
Associazioni e cardinalità delle associazioni.  
Attributi delle associazioni.  
Attributi chiave.  
Rappresentazione grafica di entità e relazioni.

**Livello base:**

Individua le entità, gli attributi e le relazioni tra entità in situazioni note.  
Rappresenta graficamente entità e relazioni di un problema.  
Progetta semplici schemi E/R.

**AREA TEMATICA: Il modello relazionale**

**COMPETENZA**

Utilizzare il modello relazionale per rappresentare un problema reale a partire da un modello E/R.

**ABILITA'**

Applicare ad un problema reale la progettazione concettuale e logica.  
Conoscere e applicare la terminologia e le definizioni relative al modello relazionale.

**CONOSCENZE**

Relazione.  
Relazioni e chiavi.  
Conversione da schema E/R a schema relazionale.  
Integrità referenziale.  
Normalizzazione: prime tre forme normali.

**Livello base:**

Traduce da schema E/R a modello relazionale.  
Applica il concetto di integrità referenziale.  
Conosce le forme normali e la loro importanza.

**AREA TEMATICA: Il linguaggio SQL**

**COMPETENZE**

Conoscere ed utilizzare gli strumenti per la gestione ed interrogazione di basi di dati.



**ABILITA'**

Applicare ad un problema reale la progettazione concettuale, logica, e fisica.  
Utilizzare il linguaggio SQL per la definizione dei dati di un problema reale.  
Utilizzare il linguaggio SQL per l'interrogazione e la manipolazione dei dati di un problema reale.

**CONOSCENZE**

Creazione di tabelle (Data Definition Language).  
Inserimento, modifica, cancellazione (Data Manipulation Language).  
Operatori relazionali (Query Language): selezione, proiezione, giunzione.  
Funzioni di aggregazione.  
Ordinamento e raggruppamenti.  
Condizioni di ricerca.

**Livello base:**

Conosce i principali comandi del linguaggio SQL (DDL, DML e QL).  
Interroga la base di dati con semplici interrogazioni.  
Crea e gestisce semplici basi di dati.

**QUINTO ANNO**

**AREA TEMATICA: Elementi di programmazione**

**COMPETENZE**

Individuare i casi in cui è utile suddividere un problema in sottoproblemi.  
Risolvere problemi progettando soluzioni che fanno uso di funzioni.

**ABILITA'**

Sa scomporre un problema in sottoproblemi facendo uso di funzioni anche ricorsive.

**CONOSCENZE**

Ricorsione e algoritmi ricorsivi.

**Livello base:**

Riconosce e sa descrivere programmi che fanno uso di funzioni ricorsive.

**AREA TEMATICA: Principi teorici della computazione**

**COMPETENZA**

Individuare i parametri di qualità di un algoritmo .

**ABILITA'**

Saper analizzare la complessità di un algoritmo.  
Saper confrontare e classificare gli algoritmi in base alle loro caratteristiche.

**CONOSCENZE**

Principi teorici della computazione: tipi di complessità computazionale di un algoritmo.  
Complessità computazionale temporale: il modello di costo.  
Analisi degli algoritmi e calcolo della complessità di un algoritmo in termini di passi base.  
Complessità asintotica e notazione O.  
Efficienza di un algoritmo.  
Classi di complessità dei problemi.

**Livello base:**

Conosce i principi teorici della computazione.  
Utilizza e descrive correttamente il concetto di complessità asintotica e notazione O-grande.



Sa analizzare semplici algoritmi in termini di passi base.  
Esprime i concetti in modo chiaro e con terminologia appropriata.

**AREA TEMATICA: Algoritmi notevoli**

**COMPETENZA**

Avere una sufficiente padronanza del linguaggio C++ per sviluppare semplici ma significative applicazioni in ambito scientifico sfruttando algoritmi noti.

**ABILITA'**

Saper implementare in C/C++ gli algoritmi studiati.  
Distinguere e analizzare, anche per efficienza, i principali algoritmi di calcolo scientifico, ordinamento e ricerca.

**CONOSCENZE**

**Algoritmi di ricerca scelti tra i seguenti:**

- Ricerca lineare
- Ricerca binaria o dicotomica

**Algoritmi di ordinamento scelti tra i seguenti:**

- Ordinamento basato sulla ricerca del minimo
- Insertion sort
- Bubble sort
- Merge sort
- Quick sort

**Algoritmi per il calcolo approssimato scelti tra i seguenti:**

- Numero e
- Radice di un'equazione con il metodo di bisezione
- Radice quadrata
- Calcolo di pi greco
- Seno di un angolo tramite sviluppo in serie di Taylor
- Aree: metodo dei rettangoli, metodo dei trapezi, metodo Monte Carlo

**Livello base:**

Comprende gli algoritmi e conosce la teoria matematica che è a loro fondamento.  
Sa applicare gli algoritmi a casi reali.

**AREA TEMATICA: Reti di computer**

**COMPETENZE**

Identificare le varie tipologie di reti e i protocolli di trasferimento adatti al tipo di dato.  
Configurare manualmente la rete di un PC.  
Saper utilizzare le principali applicazioni di rete.

**ABILITA'**

Saper classificare le tecniche di trasferimento dei dati/informazioni.  
Saper riconoscere il giusto protocollo da utilizzare nello scambio fra reti.  
Saper visualizzare e/o analizzare il traffico di rete usando appositi strumenti.

**CONOSCENZE**

Reti e protocolli.  
Modello client-server e p2p.  
Struttura e classificazione delle reti: estensione, topologia, commutazione.  
Mezzi trasmissivi.  
Modello ISO/OSI.  
Modello TCP/IP:  
Principali protocolli di applicazione: HTTP, FTP, telnet, SSH.  
Protocolli di posta elettronica: SMTP, POP3, IMAP.  
Indirizzi IP, subnetting e DNS.



Servizi di base: ifconfig, ping e traceroute.

Analisi del traffico di rete: Wireshark.

Sicurezza dei sistemi informatici.

**Livello base:**

Conosce le strutture e i tipi di classificazione usati nelle reti di computer.

Conosce l'architettura a pila ISO/OSI e TCP/IP.

Conosce e sa descrivere le responsabilità dei livelli.

Conosce e sa descrivere i principali protocolli di livello applicazione.

Conosce e sa descrivere la struttura di un indirizzo IP.

Conosce le principali problematiche di sicurezza e i tipi di attacco.