

Progettazione Disciplinare di Informatica per il Liceo Scientifico, Scienze Applicate “IIS Galilei Artiglio”

1. Introduzione

La disciplina Informatica all'interno del Liceo Scientifico, opzione Scienze Applicate, prevede un percorso quinquennale con due ore settimanali di lezione in aula.

Le linee programmatiche del Liceo Scientifico delle Scienze Applicate STEM seguiranno in linea di massima quelle del tradizionale Liceo Scientifico delle Scienze Applicate, ma con un particolare approfondimento delle materie scientifiche e un approccio pedagogico profondamente diverso e innovativo. Gli stessi contenuti disciplinari saranno affrontati attraverso una metodologia che enfatizza l'interdisciplinarietà, la pragmaticità e la sperimentazione continua, anche servendosi dei nuovi strumenti digitali, in modo da rendere gli studenti protagonisti attivi del loro percorso di apprendimento. Questo approccio integrato non solo sviluppa le conoscenze disciplinari, ma favorisce l'acquisizione di competenze trasversali e critiche per la società moderna e il mondo del lavoro contemporaneo, in linea con la natura STEM del corso di studi.

2. Obiettivi generali di apprendimento

Primo Biennio

Gli obiettivi minimi previsti per la disciplina, in termini di conoscenze, abilità e competenze, per il primo anno sono:

- Conoscere il concetto di informatica e gli ambiti di interesse e applicazione; conoscere il concetto di informazione e dato.
- Conoscere il concetto di elaboratore e tipologie e gli ambiti di impiego (calcolo, comunicazione, lavori di ufficio, controllo, intrattenimento, scientifico, ecc.).
- Conoscere la differenza tra un sistema di numerazione addizionale e posizionale e i sistemi posizionali più utilizzati in ambito informatico.
- Conoscere i concetti di peso e bit.
- Conoscere e saper utilizzare le procedure e le regole di conversione di numeri interi, relativi nei differenti sistemi di numerazione.
- Saper svolgere operazioni aritmetiche nei differenti sistemi di numerazione.
- Conoscere l'architettura di un elaboratore, le caratteristiche e il funzionamento dei componenti hardware elementari: periferiche di ingresso e di uscita; l'unità di elaborazione, i supporti di memorizzazione.
- Saper conoscere le funzioni del sistema operativo:
 - gestire l'interfaccia utente: gli elementi del desktop, le icone, il pulsante Start, la barra delle applicazioni, i file e delle cartelle.
- Conoscere il concetto di rete e saper utilizzare il software e i servizi di rete: un browser, i motori di ricerca, la posta elettronica e i cloud di archiviazione dei file online.

- Conoscere i pericoli di Internet e gli aspetti salienti della normativa relativa al diritto informatico.
- Conoscere e saper utilizzare le funzionalità principali di un software di videoscrittura (word processing/elaboratore di testo) per creare documenti di tipo differente.
- Conoscere e saper utilizzare le funzionalità principali di un foglio di calcolo: il formato campo di una cella; i comandi di formattazione del foglio e dei suoi elementi; le formule (regole sintattiche) e funzioni matematiche, logiche (di libreria) con eventuale utilizzo dei riferimenti di cella assoluti, relativi e misti.; la creazione di elenchi di dati con il riempimento automatico; la creazione di grafici.
- Conoscere le caratteristiche di una presentazione e scegliere le più appropriate (in base all'ambito di utilizzo), conoscere e saper utilizzare le funzionalità principali di un software per l'elaborazione di presentazioni per: la creazione e formattazione del layout e dei contenuti delle pagine;
- Saper utilizzare la guida in linea dei software applicativi utilizzati.
- Saper utilizzare efficacemente un servizio di posta elettronica per mandare e ricevere email
- Conoscere e saper utilizzare i software della Google Suite, con particolare riferimento a Google Classroom

Nel secondo anno, gli obiettivi si completano con:

- Conoscere il concetto di informatica e gli ambiti di interesse e applicazione; conoscere il concetto di informazione e dato.
- Conoscere il concetto di elaboratore e tipologie e gli ambiti di impiego (calcolo, comunicazione, lavori di ufficio, controllo, intrattenimento, scientifico, ecc.).
- Conoscere il concetto di software e le tipologie.
- Conoscere il concetto di codice/linguaggio e gli elementi caratteristici.
- Acquisire il concetto di Algoritmo
- conoscere il concetto di Codice sorgente e codice eseguibile
- Saper individuare e distinguere all'interno di un problema le variabili e le costanti definendole con il tipo più opportuno.
- Saper operare con le istruzioni di I/O e con le istruzioni di assegnazione.
- Saper progettare il DAB di un algoritmo e la pseudocodifica relativa all'intero algoritmo: intestazione, dichiarazione variabili e costanti, istruzioni.
- Conoscere e saper scrivere un algoritmo utilizzando le strutture di controllo più adatte a risolvere semplici problemi: selezione alternativa, selezione multipla, iterazione codificando l'algoritmo risolutivo con i linguaggi C++/Python.
- Implementazione di semplici programmi utilizzando in modo corretto la sintassi del linguaggio C++/Python scrivendo codice di indentazione, commenti.
- Saper utilizzare il software gratuito Flowgorithm per Diagrammi a Blocchi di algoritmi.
- Saper utilizzare il software gratuito DevC++ quale editor per linguaggio C++

Secondo biennio

Obiettivi minimi per il **terzo anno**:

- Acquisire il concetto di algoritmo e le sue proprietà; codice sorgente ed eseguibile
- Saper individuare e distinguere all'interno di un problema le variabili e le costanti, definendole con il tipo più opportuno
- Saper operare con istruzioni di I/O e con le istruzioni di assegnamento
- Conoscere e saper scrivere un algoritmo utilizzando le strutture di controllo più adatte a risolvere semplici problemi:
 - selezione binaria (if/else), selezione unaria (if)
 - Iterazione: precondizionale (while), postcondizionale (do/while), determinata (for)
- Essere in grado di scrivere programmi scomposti in funzioni/procedure; dichiarare ed invocare funzioni, individuando i parametri, effettuando il passaggio di parametri per valore o per riferimento e restituendo il corretto output
- Conoscere il concetto di visibilità di una variabile locale e globale
- Saper organizzare i dati in array; implementare algoritmi sugli array, anche con l'uso delle funzioni, per la gestione di semplici operazioni con essi: caricamento; visualizzazione; conteggio, somma degli elementi che soddisfano una determinata condizione; calcolo media degli elementi (o degli elementi che soddisfano una determinata condizione); ricerca elemento min/max
- Saper organizzare i dati in matrici e implementare semplici algoritmi, anche con l'uso delle funzioni, per la gestione di operazioni con esse: caricamento e visualizzazione; visualizzazione e/o somma degli elementi di una specifica riga/colonna o diagonale principale (per matrici quadrate)
- Saper organizzare i dati in struct e implementare semplici algoritmi, anche con l'uso delle funzioni, per la gestione di operazioni con esse: scrittura, lettura e modifica di un campo di una struct
- Saper discriminare la struttura dati più corretta a seconda del problema posto
- Implementare programmi utilizzando in modo corretto la sintassi di più linguaggi scrivendo codice con indentazione, commenti

Obiettivi minimi per il **quarto anno**:

- Conoscere i concetti di base del paradigma di programmazione orientata agli oggetti: classe, oggetto, attributi, metodi; metodo costruttore; ereditarietà
- Conoscere le regole di visibilità: public, private, protected
- Conoscere il concetto di polimorfismo e saper ridefinire un metodo: overloading, overriding
- Saper progettare e implementare semplici classi concrete o una gerarchia di classi

Monoennio finale

Obiettivi minimi per il **quinto anno**:

- Conoscere concetti generali sulle basi di dati: caratteristiche di una base di dati e funzionalità di un DBMS, utenti del DBMS
- Saper individuare gli elementi essenziali della "realtà di interesse" nella progettazione di una base di dati e saperli realizzare attraverso il modello concettuale, modello E/R: entità, attributi, chiave primaria, associazioni e cardinalità
- Saper produrre uno schema concettuale ben progettato, consistente e già normalizzato
- Essere in grado di applicare le regole di derivazione per tradurre uno schema concettuale in uno schema logico relazionale; conoscere il concetto di chiave esterna
- Saper utilizzare le istruzioni fondamentali del linguaggio SQL per creare, manipolare e interrogare un database: SELECT, equi-join, condizioni di ricerca, funzioni di aggregazione, semplici raggruppamenti, condizioni sui raggruppamenti
- Comprendere i concetti riguardanti le reti: scopi, componenti fondamentali e classificazioni.
- Conoscere i modelli di riferimento ISO/OSI e TCP/IP.
- Crittografia e algoritmi crittografici.