

PROGRAMMAZIONE DI INFORMATICA

Liceo Scientifico scienze applicate "P. Gobetti"

Per ciascuna classe, la programmazione mette in evidenza, per ogni nucleo, gli argomenti trattati che si ritengono **essenziali**, in accordo con le indicazioni nazionali.

Le abilità indicate nella seconda colonna sono gli obiettivi raggiungibili assimilando i saperi della prima colonna.

I **contenuti minimi** (nella terza colonna) indicano invece le richieste minime per il livello di sufficienza.

Nella tabella sotto (priva di obiettivi minimi) sono invece riportati i principali approfondimenti non sempre trattati.

CLASSI PRIME

CONOSCENZE	ABILITA'	OBIETTIVI MINIMI
La rappresentazione dell'informazione <ul style="list-style-type: none"> • Le conversioni tra basi • La codifica delle immagini • Il codice ASCII 	<ul style="list-style-type: none"> • Saper convertire tra le basi 2, 16, 10 • Saper descrivere come viene codificata l'informazione 	<ul style="list-style-type: none"> • Saper convertire tra le basi 2, 16, 10
La posta elettronica e la netiquette <ul style="list-style-type: none"> • Utilizzo della posta elettronica • La netiquette 	<ul style="list-style-type: none"> • Saper mandare mail gestendo correttamente destinatari e allegati • Conoscere i principi della netiquette 	<ul style="list-style-type: none"> • Saper mandare mail gestendo correttamente destinatari e allegati • Conoscere i principi della netiquette
Il foglio di calcolo <ul style="list-style-type: none"> • Il foglio di calcolo: organizzazione dei dati, formule, funzioni, grafici 	<ul style="list-style-type: none"> • Saper organizzare dati su un foglio di calcolo • Saper utilizzare le principali formule e funzioni del foglio di calcolo • Saper creare grafici a partire dai dati 	<ul style="list-style-type: none"> • Saper organizzare dati su un foglio di calcolo • Saper utilizzare le principali formule e funzioni del foglio di calcolo • Saper creare grafici a partire dai dati
Gli algoritmi <ul style="list-style-type: none"> • Il concetto di algoritmo 	<ul style="list-style-type: none"> • Saper codificare in diagramma di flusso o pseudocodice il processo risolutivo di un problema 	<ul style="list-style-type: none"> • Saper eseguire una procedura descritta in un diagramma di flusso o pseudocodice

<ul style="list-style-type: none"> • Diagrammi di flusso e/o pseudocodice 		
--	--	--

APPROFONDIMENTI NON SEMPRE TRATTATI		
L'hardware e il software		
<ul style="list-style-type: none"> • Principali componenti del pc e loro funzione • Licenze d'uso • Codice sorgente 		
I software di office automation		
<ul style="list-style-type: none"> • I software di videoscrittura • I software di presentazione 		
Storia dell'Informatica		

CLASSI SECONDE

CONOSCENZE	ABILITA'	OBIETTIVI MINIMI
Le basi della programmazione imperativa <ul style="list-style-type: none"> • I tipi e la dichiarazione delle variabili • La sequenza • La selezione • L'iterazione 	<ul style="list-style-type: none"> • Conoscere i principali tipi primitivi delle variabili • Saper utilizzare la sequenza, la selezione e l'iterazione per implementare semplici programmi in C/C++ 	<ul style="list-style-type: none"> • Conoscere i principali tipi primitivi delle variabili • Saper utilizzare la sequenza, la selezione e l'iterazione per implementare semplici programmi in C/C++
Il web e le reti <ul style="list-style-type: none"> • La struttura di una rete • Cenni sull'architettura client-server • Gli indirizzi IP • Sicurezza in rete 	<ul style="list-style-type: none"> • Conoscere i principi del funzionamento della rete internet • Saper navigare in modo sicuro • Conoscere la struttura degli indirizzi IP e dei domini web 	<ul style="list-style-type: none"> • Conoscere la terminologia della rete internet • Saper navigare in modo sicuro

APPROFONDIMENTI NON SEMPRE TRATTATI		
--	--	--

Software di creazione di presentazioni animate <ul style="list-style-type: none"> • Powtoon
Alcuni servizi della Google Suite <ul style="list-style-type: none"> • Maps • Google Moduli

CLASSI TERZE

CONOSCENZE	ABILITA'	OBIETTIVI MINIMI
Programmazione C/C++ <ul style="list-style-type: none"> • Sottoprogrammi (funzioni e procedure) • La ricorsione • Vettori • Matrici • Stringhe 	<ul style="list-style-type: none"> • Saper utilizzare i sottoprogrammi per implementare programmi in C/C++ • Conoscere l'uso dei vettori, delle matrici e delle stringhe 	<ul style="list-style-type: none"> • Saper utilizzare i sottoprogrammi per implementare programmi in C/C++ • Conoscere l'uso dei vettori
I linguaggi del web <ul style="list-style-type: none"> • HTML • CSS 	<ul style="list-style-type: none"> • Saper costruire siti web in HTML, inserendo collegamenti ipertestuali 	<ul style="list-style-type: none"> • Saper costruire semplici pagine web in HTML, utilizzando gli opportuni tag

APPROFONDIMENTI NON SEMPRE TRATTATI
Piattaforme per la costruzione di siti web <ul style="list-style-type: none"> • Piattaforme What You See Is What You Get (Wordpress, Google Sites...)

CLASSI QUARTE

Saperi essenziali	Abilità	Contenuti minimi per la sufficienza
Il modello ER <ul style="list-style-type: none"> • Diagrammi ER: Entità e attributi, associazioni • Dal diagramma ER al database 	<ul style="list-style-type: none"> • Saper ricavare il diagramma ER a partire da una realtà descritta a parole • Saper progettare un DB a partire dal diagramma ER 	<ul style="list-style-type: none"> • Saper leggere un diagramma ER individuando entità, attributi, associazioni
Database <ul style="list-style-type: none"> • Linguaggio SQL: istruzioni DDL, DML e DCL • Le query: l'istruzione select 	<ul style="list-style-type: none"> • Conoscere la differenza tra le istruzioni DDL, DML e DCL • Conoscere le principali istruzioni del linguaggio SQL • Saper costruire una query in SQL 	<ul style="list-style-type: none"> • Saper distinguere le varie istruzioni DML (insert, delete, update) • Saper costruire una query in SQL

APPROFONDIMENTI NON SEMPRE TRATTATI

Programmazione orientata agli oggetti in C++

CLASSI QUINTE

CONOSCENZE	ABILITA'	OBIETTIVI MINIMI
Calcolo numerico <ul style="list-style-type: none"> • Metodi di approssimazione (alcuni, a scelta del docente: approssimazione dell'integrale, ricerca degli zeri di una funzione, interpolazione...) 	<ul style="list-style-type: none"> • Conoscere alcune applicazioni del calcolo numerico nei metodi di approssimazione • Sapere eseguire le procedure di calcolo numerico trattate 	<ul style="list-style-type: none"> • Conoscere alcune applicazioni del calcolo numerico nei metodi di approssimazione • Saper descrivere le procedure di calcolo numerico trattate
Le reti <ul style="list-style-type: none"> • Il modello ISO/OSI e i suoi livelli • Il modello TCP/IP e gli indirizzi IP • Topologia delle reti • Classificazione delle reti sulla base dell'estensione • Architettura client-server e peer-to-peer 	<ul style="list-style-type: none"> • Conoscere l'architettura del modello ISO/OSI e i suoi livelli • Conoscere il modello TCP/IP e saperlo confrontare con l'ISO/OSI • Conoscere alcuni protocolli del modello TCP/IP 	<ul style="list-style-type: none"> • Conoscere il modello TCP/IP • Conoscere alcuni protocolli del modello TCP/IP • Conoscere il funzionamento degli indirizzi IP e dei principali dispositivi di rete

<ul style="list-style-type: none"> • Principali dispositivi di rete (modem, router...) 	<ul style="list-style-type: none"> • Conoscere il funzionamento degli indirizzi IP e dei principali dispositivi di rete 	
Crittografia <ul style="list-style-type: none"> • Breve storia della crittografia • Codici mono e polialfabetici • Crittografia a chiave simmetrica e asimmetrica (RSA) • La macchina Enigma e Colossus, Turing 	<ul style="list-style-type: none"> • Conoscere le principali tappe della storia della crittografia • Conoscere alcuni esempi di crittografia e saperli utilizzare • Saper confrontare metodi diversi 	<ul style="list-style-type: none"> • Conoscere le principali tappe della storia della crittografia • Conoscere alcuni esempi di crittografia
Complessità computazionale <ul style="list-style-type: none"> • Gli ordini di complessità computazionale • Costo di un algoritmo • Confronto tra complessità di algoritmi notevoli 	<ul style="list-style-type: none"> • Conoscere il concetto di complessità computazionale • Saper quantificare il costo di un algoritmo • Saper confrontare algoritmi diversi 	<ul style="list-style-type: none"> • Conoscere il concetto di complessità computazionale • Saper quantificare il costo di un algoritmo

APPROFONDIMENTI NON SEMPRE TRATTATI

Calcolo matriciale

- Le matrici: classificazione e determinante
- Somma e prodotto matriciale
- Rango e applicazioni in algebra lineare

Sistemi e automi

- I sistemi
- Automi a stati finiti
- La macchina di Turing