
Opzione complementare di informatica

A. Obiettivi generali

L'informatica è sempre più presente nella vita quotidiana. L'insieme delle discipline scientifiche è influenzato dalle sue applicazioni. L'opzione complementare sviluppa nello studente le attitudini che gli permettono di apprezzare la natura e l'importanza dell'informatica, così come di valutarne le sue applicazioni particolari.

L'informatica mette in relazione il pensiero scientifico, la matematica e la tecnica in una sola disciplina. Il lavoro di gruppo orientato alla realizzazione di progetti, l'elaborazione di differenti soluzioni e il paragone critico sono elementi fondamentali della formazione in questo ambito.

L'opzione complementare di informatica sviluppa la capacità di analisi e di modellazione dei problemi, così come la concezione di soluzioni algoritmiche e non.

La qualità delle soluzioni proposte è valutabile direttamente grazie allo sviluppo delle soluzioni sotto forma di programmi o combinazioni di applicazioni.

Lo studente scopre quali soluzioni sono tecnicamente realizzabili e quali risorse sono necessarie.

L'opzione complementare di informatica fornisce le basi di algoritmica, di programmazione e di gestione dell'informazione in rete. La realizzazione di progetti permette di approfondire questi temi.

B. Considerazioni e spiegazioni

L'apprendimento dell'informatica parte dalla scoperta dei principi di base del trattamento dell'informazione, si occupa della formalizzazione e della modellazione dei problemi e permette di arrivare ad una realizzazione pratica così come ad una valutazione del risultato raggiunto.

Non si tratta quindi semplicemente di saper utilizzare le nuove tecnologie dell'informazione e della comunicazione ma è essenziale comprendere i principi del trattamento dell'informazione per concepire e utilizzare per la risoluzione di specifici problemi nell'ambito di svariati domini.

I fondamenti dell'informatica mettono in evidenza la frontiera tra problemi che possono essere affrontati in modo algoritmico o meno, così come le risorse necessarie e sufficienti. In pratica si evidenziano le potenzialità del calcolatore e le sue risorse fisiche, con l'obiettivo di una risoluzione di problemi concreti.

Con questi fondamenti si possono affrontare e concepire soluzioni a problemi di differenti ambiti non restringendo il campo alla scienza esatta e alla matematica soltanto, ma applicandole – in modo creativo – a tutte le discipline che richiedono di un trattamento dell'informazione.

Saper programmare implica la conoscenza di un linguaggio formale per indirizzare istruzioni alle componenti dell'elaboratore. In questo modo si rende competente lo studente alla soluzione pratica di problemi di natura algoritmica.

L'aspetto interdisciplinare, legato alla combinazione di aspetti tecnici e matematici per la soluzione di un problema tratto da altri ambiti, rende l'informatica una disciplina complessa e che richiede di un alto grado di astrazione. In questo senso l'informatica fornisce un importante contributo alla formazione di una cultura scientifica moderna e alla cultura generale.

La sperimentazione pratica risulta così fondamentale e di alto valore didattico e formativo. La possibilità di sviluppare e adattare continuamente il proprio prodotto alle specifiche e alle necessità rende particolarmente motivante il processo di apprendimento. Per la trasmissione di tali esperienze orientate alla risoluzione di problemi pratici si prestano piccoli progetti mirati, da realizzare in gruppo.

C. Obiettivi fondamentali

Conoscenze

- Comprendere le nozioni di base e i concetti fondamentali necessari alla modellazione, all'analisi dei problemi e alla concezione di soluzioni informatiche.
- Conoscere le procedure di valutazione e la pertinenza delle soluzioni possibili.
- Conoscere la base di un linguaggio di programmazione.
- Conoscere le differenti rappresentazioni dell'informazione.
- Comprendere le basi della comunicazione digitale.
- Conoscere i limiti delle applicazioni informatiche.

Saper fare

- Analizzare e strutturare un problema dato tratto da ambiti di applicazione differenti.
- Elaborare, valutare algoritmi e codificarli in un linguaggio di programmazione.
- Mettere a punto modelli di informazione adatti alle specifiche di un problema.
- Valutare e documentare la pertinenza, l'efficacia, l'efficienza e l'usabilità di una soluzione informatica.

Attitudini

- Dar prova di spirito critico nell'analisi e nella valutazione di soluzioni informatiche.
- Essere disponibile a lavorare su progetti e in gruppo così come agli scambi interdisciplinari.
- Pianificare e agire in modo strutturato.
- Mostrare perseveranza nella ricerca e nell'implementazione di soluzioni informatiche.
- Confrontarsi con le implicazioni dell'informatica nella vita quotidiana.

D. Contenuti

Il piano di studio è composto da tre moduli di base, prioritari, e da un gruppo di moduli facoltativi di approfondimento. Nel corso del biennio possono essere trattati, a dipendenza del livello di dettaglio, alcuni moduli facoltativi, possibilmente in campi diversi.

Cenni sulla storia dell'informatica devono essere integrati in tutti i moduli.

Quasi tutti i moduli contengono delle parti pratiche importanti: per questo motivo le classi dovranno essere composte da un numero ragionevole di allievi e ogni allievo dovrà avere a disposizione un computer, in modo da poter lavorare attivamente e autonomamente.

Temi di base

<i>Programmazione, strutture di dati e algoritmi</i>	<i>Argomenti</i> <ul style="list-style-type: none">• Evoluzione dei computer, dei linguaggi di programmazione e dei paradigmi di programmazione.• Cenni di architettura di un computer.• Pseudocodice, compilatori ed interpreti.• Introduzione di un linguaggio di programmazione, scelto dal docente, che permetta di implementare soluzioni a problemi posti da ambiti diversi (matematica, fisica, chimica, biologia, economia, ...): dato, programma, variabile
--	---

	<p>(locale e globale), costante, parametro, istruzione (di assegnazione, condizionale, di ripetizione, ...), sottoprogramma (funzione, procedura, ...), librerie, ...</p> <ul style="list-style-type: none"> • Le principali fasi della programmazione: descrizione del problema, scomposizione del problema, codifica, test, correzione degli errori, verifica, ottimizzazione, documentazione. • Strutture di dati fondamentali: <i>array</i>, liste concatenate, code, alberi, grafi, insiemi. • Algoritmi di ordinamento e algoritmi di ricerca: studio, implementazione, cenni sulla correttezza e sulla complessità computazionale. <p><i>Obiettivi</i> Sviluppare l'attitudine al pensiero logico e astratto; essere in grado di analizzare semplici problemi ed algoritmi ed implementarli in un linguaggio di programmazione; capire che la programmazione non è "tenta e correggi gli errori" ma è composta da diverse fasi, tutte importanti; saper scegliere e riconoscere le strutture di dati adeguate all'implementazione di algoritmi per la soluzione di problemi elementari; conoscere i principali algoritmi di ordinamento e di ricerca.</p>
<i>Internet</i>	<p><i>Argomenti</i></p> <ul style="list-style-type: none"> • Aspetti storici e sociali di Internet. • Funzionamento del <i>browser</i>, il principio <i>server-client</i>. • Iper testi e documenti multimediali, costruzione di pagine HTML. • I principali protocolli di comunicazione (HTTP, TCP/IP, FTP, TELNET, ...) e tipologia delle reti. • La sicurezza: possibili minacce, forme di attacco, protezioni; gli aspetti giuridici. • I servizi Web: <i>Webhosting</i>, <i>Content Management</i>, <i>streaming</i> audio e video, ... <p><i>Obiettivi</i> Apprendere il percorso relativo alla nascita di Internet; formare uno spirito critico relativo agli aspetti culturali, sociali ed etici di Internet; essere in grado di scrivere e pubblicare pagine Web; conoscere i principali protocolli in uso su Internet; conoscere gli aspetti relativi alla sicurezza.</p>
<i>Sistemi informativi</i>	<p><i>Argomenti</i></p> <ul style="list-style-type: none"> • Introduzione alle strutture di dati dinamiche fondamentali per la gestione dei dati (<i>B-Tree</i>, <i>hashing</i>, ...) e alle operazioni ad esse correlate (ricerca, inserimento di un elemento, cancellazione di un elemento). • Introduzione alle basi di dati: relazionali e non (relazioni, attributi, valori, chiavi, forme normali, cenni di SQL, ...). • Interrogazione e manipolazione di una base di dati. • Analisi, progettazione e costruzione di semplici basi di dati. • Algoritmi di ricerca in Internet: introduzione alla ricerca, valutazione di una ricerca, strategie fondamentali di una ricerca, metodi statistici del <i>Page ranking</i>. <p><i>Obiettivi</i> Conoscere le basi teoriche per l'analisi dei dati; saper progettare, costruire e interrogare semplici basi di dati; saper svolgere ricerche avanzate e critiche su Internet.</p>

Proposte di temi di approfondimento

<p><i>Programmazione: introduzione alla programmazione orientata agli oggetti</i></p>	<p>Argomenti</p> <ul style="list-style-type: none">• Il paradigma della programmazione orientata agli oggetti, cenni sull'ingegneria del <i>software</i>.• Il linguaggio UML per la modellazione dei sistemi (casi di uso, diagrammi delle classi, di sequenza di azione e di stato)• Oggetti e classi, istanze, metodi.• Modularità, ereditarietà, polimorfismo, incapsulamento dei dati, occultamento dei dati, interfacce.• Riutilizzabilità del <i>software</i>.• Sviluppo di un piccolo progetto (lavoro a gruppi). <p>Obiettivi</p> <p>Apprendere il paradigma della programmazione orientata agli oggetti, saper sviluppare piccoli progetti utilizzando un linguaggio di programmazione orientato agli oggetti o appositi <i>tool</i>.</p>
<p><i>Programmazione: metodi matematici</i></p>	<p>Argomenti</p> <ul style="list-style-type: none">• Algoritmi dell'analisi numerica: ricerca di uno zero di una funzione, approssimazione polinomiale, metodi di interpolazione, derivazione numerica, integrazione numerica, risoluzione numerica di equazioni differenziali, ...• Algoritmi dell'algebra lineare: sistemi di equazioni e disequazioni lineari, ottimizzazione discreta, calcolo di determinanti, inversione di matrici, calcolo di autovalori e autovettori, ...• Implementazione degli algoritmi mediante un linguaggio di programmazione oppure mediante apposito software matematico (Octave, Scilab, Matlab, ...). <p>Obiettivi</p> <p>Essere in grado di implementare correttamente algoritmi numerici e saperne valutare la correttezza e il costo computazionale.</p>
<p><i>Programmazione di robot</i></p>	<p>Argomenti</p> <ul style="list-style-type: none">• Interfaccia <i>hardware/software</i>.• Coordinate omogenee.• Programmazione dei movimenti del robot.• Sviluppo di strategie di comportamento (per esempio: uso di sensori per riconoscere un ostacolo, per distinguere il chiaro dallo scuro, ...). <p>Obiettivi</p> <p>Definire le azioni elementari di un robot e saperle implementare nell'apposito linguaggio di interfaccia; programmare un robot che sia in grado di svolgere dei compiti elementari.</p>
<p><i>Sviluppo di algoritmi ed analisi dell'efficienza</i></p>	<p>Argomenti</p> <ul style="list-style-type: none">• Algoritmi di ottimizzazione di tipo <i>greedy</i>.• <i>Divide et impera</i> come metodo ricorsivo.• Programmazione dinamica come metodo iterativo.• Ricerca locale nell'ottimizzazione.• <i>Backtracking</i> come metodo di ricerca esaustiva.• Implementazione degli algoritmi in un linguaggio di programmazione.

	<p><i>Obiettivi</i> Apprendere alcune tecniche algoritmiche classiche e saperle utilizzare nella risoluzione di semplici problemi; saper implementare algoritmi avanzati in un linguaggio di programmazione; saper riconoscere l'efficienza degli algoritmi.</p>
<i>Algoritmi geometrici</i>	<p><i>Argomenti</i></p> <ul style="list-style-type: none"> • Ricerca di due punti aventi distanza minima. • Intersezione di oggetti geometrici. • Involucro convesso. • Scomposizione in triangoli di una figura piana. • Problemi di visibilità. <p><i>Obiettivi</i> Apprendere soluzioni algoritmiche a semplici problemi geometrici.</p>
<i>Architettura di un computer</i>	<p><i>Argomenti</i></p> <ul style="list-style-type: none"> • Macchina di von Neumann. • Organizzazione di un computer: processore, memoria, bus, input, output. • Dispositivi di memorizzazione, di input e output. • Aritmetica del calcolatore (basi 2, 8 e 16). • Discretizzazione grafica (schermo, stampante, ...). <p><i>Obiettivi</i> Comprendere il modello architetturale di von Neumann e il funzionamento di un computer; sapere in quale forma viene trattata l'informazione in un computer; conoscere il significato delle unità di misura tipiche (bit, byte, ..., pixel, dpi, ...).</p>
<i>Crittologia</i>	<p><i>Argomenti</i></p> <ul style="list-style-type: none"> • Concetti fondamentali: mittente, destinatario, testo in chiaro, testo cifrato, chiave di cifratura, chiave di decifratura. • Crittologia = Crittografia + Crittoanalisi. • Metodi crittografici classici (storia della crittografia, metodo di Cesare, metodo di Vigenère, principio di Kerckhoff, ...). • Crittosistemi simmetrici: punti forti e deboli, attacchi ai sistemi monoalfabetici, <i>One Time Pad</i>, DES, ... • Crittografia a chiave pubblica: le funzioni unidirezionali, il crittosistema RSA, firme digitali, commercio elettronico, ... • Sicurezza: autenticità, identificazione, integrità, autorizzazioni, problemi giuridici, ... <p><i>Obiettivi</i> Conoscere lo sviluppo storico dei metodi crittografici; riconoscere punti di forza e punti deboli dei crittosistemi studiati; apprendere i metodi che permettono di attuare una comunicazione sicura; saper riconoscere i rischi di una comunicazione.</p>
<i>Automati</i>	<p><i>Argomenti</i></p> <ul style="list-style-type: none"> • Concetti fondamentali: alfabeto, simboli, stringhe, linguaggi, sintassi, semantica, relazione, problemi decisionali. • Il concetto di automa a stati finiti come semplice modello di computer: configurazione, passo di calcolo, calcolo. • Metodi per progettare automi. • Verifica della correttezza mediante induzione completa.

	<ul style="list-style-type: none"> • Progettazione di un automa per il comando di un ascensore e di un semaforo. • Dimostrazione della non esistenza di automi per la risoluzione di certi problemi. <p><i>Obiettivi</i> Essere in grado di costruire semplici automi; comprendere il problema della calcolabilità.</p>
<i>Computer Graphics</i>	<p><i>Argomenti</i></p> <ul style="list-style-type: none"> • Sistemi di coordinate e trasformazioni affini nello spazio e nel piano. • Rappresentazione di oggetti geometrici del piano (punti, rette, cerchi, ellissi, curve di Bézier, ...) e dello spazio (punti, rette, cubi, sfere, coni, cilindri, ...) e modellazione geometrica. • Grafica <i>bitmap</i> e grafica vettoriale. • Modelli di colori (RGB, CMYK). <p><i>Obiettivi</i> Comprendere come vengono visualizzati e manipolati oggetti 2D e 3D da un computer; conoscere le caratteristiche delle diverse rappresentazioni grafiche; comprendere come vengono codificati i colori; risolvere semplici problemi di <i>computer graphics</i> mediante l'uso di librerie grafiche o apposito <i>software</i>.</p>
<i>Multimedia</i>	<p><i>Argomenti</i></p> <ul style="list-style-type: none"> • Formati grafici: tipi, tecniche di compressione. • Formati audio: tipi, tecniche di compressione. • Formati video: tipi, tecniche di compressione. • Internet e il multimedia: servizi di <i>streaming</i>, implicazioni sociali e giuridiche sull'uso di contenuti multimediali. • Formati audio: tipi, tecniche di compressione. <p><i>Obiettivi</i> Riconoscere e saper trattare i diversi formati grafici, audio e video.</p>
<i>Web 2.0</i>	<p><i>Argomenti</i></p> <ul style="list-style-type: none"> • Progettazione di siti dinamici: PHP, CSS, Javascript. • Principi di base di Ajax. • Esempi e sviluppo di applicazioni Web 2.0. • XML e il Web semantico. <p><i>Obiettivi</i> Apprendere i concetti base del Web 2.0; saper creare piccole applicazioni Web 2.0.</p>

E. Valutazione

Il raggiungimento degli obiettivi è verificato attraverso:

- la capacità di recepire, di riprodurre e sviluppare autonomamente i principali aspetti dei temi trattati;
- esercitazioni in classe;
- compiti da svolgere a casa;
- presentazione di lavori di approfondimento personale o a gruppi;
- partecipazione attiva dell'allievo alle lezioni;
- lavori scritti e lavori al computer.