

13. Informatica - Opzione complementare

13.1. Obiettivi generali

L'opzione complementare di informatica approfondisce i contenuti della disciplina obbligatoria che studia l'informatica come scienza. Suo scopo principale è quello di consolidare le basi di algoritmica, di programmazione e di gestione dell'informazione in rete, ma offre anche l'occasione per introdurre elementi di robotica, intelligenza artificiale, machine learning, crittologia (e altro ancora), che sono di importanza cruciale negli sviluppi più recenti della disciplina. Proseguendo coerentemente il percorso avviato nel primo biennio di informatica obbligatoria, poi, stimola lo studente a costruire un legame il più possibile solido, continuo e proficuo tra le conoscenze relative alle basi tecnologiche della scienza informatica e le conoscenze d'ordine etico, culturale, sociale ed economico ad esse correlate. L'opzione complementare sviluppa, dunque, nello studente le attitudini che gli permettono di apprezzare la natura e le potenzialità dell'informatica, così come di valutarne le applicazioni particolari nel loro impatto sulla società e sulla cultura.

13.2. Considerazioni e spiegazioni

L'apprendimento dell'opzione complementare di informatica parte dai principi di base del trattamento dell'informazione, si occupa della formalizzazione e della modellizzazione dei problemi e permette di arrivare ad una realizzazione pratica così come ad una valutazione del risultato raggiunto. Introduce a un metodo che indica in modo chiaro i confini tra problemi concreti risolvibili algebricamente mediante processi automatizzati e problemi che non lo sono, così come le risorse necessarie di tutto il sistema informatico.

Su tali basi si possono affrontare e concepire soluzioni a problemi d'ambito diverso non limitandosi all'ambito delle scienze esatte e della matematica soltanto, ma applicandole in modo creativo a tutte le discipline che richiedono un trattamento dell'informazione.

La programmazione che serve a innescare il processo automatico che porta alla soluzione pratica di problemi di natura algoritmica implica la conoscenza di un linguaggio formale che indirizzi istruzioni alle componenti dell'elaboratore.

L'aspetto interdisciplinare legato alla combinazione di aspetti teorici e pratici tecnici e matematici, che può sollevare questioni etico-filosofiche e giuridiche nella valutazione di limiti e vantaggi, rende l'informatica una disciplina complessa che richiede un alto grado di astrazione. In questo senso l'informatica fornisce un importante contributo alla formazione di una cultura scientifica moderna.

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

L'opzione complementare di informatica, partendo dai principi di base del trattamento dell'informazione, si occupa della formalizzazione e della modellizzazione dei problemi per introdurre l'allievo a un metodo che indica in modo chiaro i confini tra problemi concreti risolvibili algebricamente mediante processi automatizzati e problemi che non lo sono e insieme gli consente di individuare le risorse necessarie di tutto il *sistema informatico*; lo scopo ultimo è quello di arrivare ad una realizzazione pratica sotto forma di programma (dunque di codice scritto in un determinato linguaggio formale che indirizza istruzioni alle componenti dell'elaboratore) e alla sua valutazione. Su tali basi si possono affrontare e concepire soluzioni a problemi d'ambito diverso non limitandosi all'ambito delle scienze esatte e della matematica soltanto, ma applicandole in modo creativo a tutte le discipline che richiedono un trattamento dell'informazione.

In quest'ottica, la sperimentazione orientata alla risoluzione di problemi pratici, individualmente o in gruppo, risulta fondamentale e di alto valore didattico e formativo; e la possibilità di sviluppare e adattare continuamente il proprio prodotto alle esigenze del sistema informatico e agli obiettivi prefissati rende particolarmente motivante il processo di apprendimento.

L'aspetto interdisciplinare legato alla combinazione di aspetti teorici e pratici tecnici e matematici, che può sollevare questioni etico-filosofiche e giuridiche nella valutazione di limiti e vantaggi, rende l'informatica una disciplina complessa che richiede un alto grado di astrazione. In questo senso l'informatica fornisce un importante contributo alla formazione di una cultura scientifica moderna.

13.3. Obiettivi fondamentali

Conoscenze

- Comprendere le nozioni di base e i concetti fondamentali necessari alla modellizzazione, 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 al lavoro 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.

13.4. Contenuti

Il piano di studio è composto da tre moduli di base, prioritari, e da un gruppo di moduli facoltativi di approfondimento, che nel corso del biennio andranno scelti possibilmente in campi diversi.

Cenni sull'impatto dell'informatica nella società 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.

Gli "obiettivi" che figurano in corsivo sono già presenti nel programma del primo biennio e riguardano aspetti non solo tecnici dei temi proposti.

Temi di base

<i>Programmazione, strutture di dati e algoritmi</i>	<p><i>Argomenti</i></p> <ul style="list-style-type: none">– Ripresa dei concetti di base relativi alla programmazione (input/output, variabili, istruzioni di assegnazione, istruzioni condizionali, istruzioni di ripetizione; visibilità delle variabili).– Sottoprogrammi (funzioni, procedure, ...) e librerie; la ricorsione.– 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.– <i>Multiprocessing</i>.– Il paradigma della programmazione orientata agli oggetti, cenni sull'ingegneria del <i>software</i>.– Oggetti e classi, istanze, metodi.– Modularità, ereditarietà, polimorfismo, incapsulamento dei dati, occultamento dei dati, interfacce.– Riutilizzabilità del <i>software</i>.– Algoritmi di ordinamento e algoritmi di ricerca: studio, implementazione, cenni sulla correttezza e sulla complessità computazionale.– 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></p> <p>Sviluppare l'attitudine al pensiero logico e astratto; 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> <p>Apprendere alcune tecniche algoritmiche classiche e saperle utilizzare nella risoluzione di problemi semplici; saper implementare algoritmi avanzati in un linguaggio di programmazione; saper riconoscere l'efficienza degli algoritmi. Approfondire lo studio di un linguaggio di programmazione, 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>Essere coscienti del fatto che nella nostra società molte decisioni sono influenzate dall'uso di algoritmi e che gli algoritmi non sono neutrali, ma riflettono precise intenzioni di chi li istruisce.</i></p>
<i>Reti e Web 2.0</i>	<p><i>Argomenti</i></p> <ul style="list-style-type: none">– I principali protocolli di comunicazione (HTTP, TCP/IP, FTP, TELNET, ...) e tipologia delle reti.– Ripresa e consolidamento di HTML.– I fogli di stile CSS.– Costruzione di pagine dinamiche (per esempio con Javascript, PHP o Python).– Interfacciamento con una base di dati.– Esempi e sviluppo di applicazioni Web 2.0; aspetti giuridici.– Programmazione (semplificata) di <i>app</i> per <i>smartphone/tablet</i> (per esempio con il software MIT appInventor).

	<p><i>Obiettivi</i></p> <p>Essere in grado di scrivere e pubblicare pagine Web, anche dinamiche; conoscere i principali protocolli in uso su Internet; conoscere le principali norme che regolano la proprietà intellettuale e la <i>privacy</i>, apprendere i concetti base del Web 2.0; saper creare piccole applicazioni Web 2.0.</p>
<i>Sistemi informativi</i>	<p><i>Argomenti</i></p> <ul style="list-style-type: none"> – Introduzione alle basi di dati: 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></p> <p>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> <p><i>Conoscere i rischi legati all'uso improprio di grandi basi di dati nel contesto economico e sociale.</i></p> <p><i>Essere consapevoli dei problemi legati alla conservazione a lungo termine delle basi di dati degli archivi digitali.</i></p>

Proposte di temi di approfondimento

Alcuni di questi temi possono essere proposti e approfonditi in un lavoro di maturità.

<i>Programmazione di robot</i>	<p><i>Argomenti</i></p> <ul style="list-style-type: none"> – Identificare le diverse componenti di un robot e capire a cosa servono. – Sensori e attuatori; passaggio dall'analogico al digitale. – Descrizione delle operazioni elementari che un robot deve svolgere per poter risolvere un problema; implementazione delle stesse in un linguaggio di programmazione. – Interfacciamento con schede Arduino o con Rasperry Pi (o simili). – Sviluppo di strategie di comportamento (per esempio: uso di sensori per riconoscere un ostacolo, per distinguere il chiaro dallo scuro, ...). – Robotica e società. <p><i>Obiettivi</i></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>Riflettere criticamente su vantaggi, limiti e prospettive della robotizzazione.</i></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>, AES, ... – Crittografia a chiave pubblica: le funzioni unidirezionali, il crittosistema RSA, firme digitali, commercio elettronico, ...

	<ul style="list-style-type: none"> – 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> <p><i>Acquisire consapevolezza delle problematiche legate alla firma e all'identità digitale.</i></p>
Automati a stati finiti	<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. – 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>
Computer Graphics	<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 modellizzazione geometrica. – Grafica <i>bitmap</i> e grafica vettoriale. – Modelli di colori (RGB, CMYK). – Programmazione grafica (per esempio con OpenGL). <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>
Blockchain	<p><i>Argomenti</i></p> <ul style="list-style-type: none"> – La tecnologia <i>blockchain</i>. – Le criptovalute. – Sicurezza. – Consumo energetico. <p><i>Obiettivi</i> Comprendere come funziona la <i>blockchain</i>; studiare applicazioni delle <i>blockchain</i> (criptovalute, ma non solo); affrontare i problemi relativi alla sicurezza.</p> <p><i>Acquisire consapevolezza delle opportunità relative ad una maggiore trasparenza economica eticamente rilevante (tracciabilità), dei problemi relativi al consumo energetico e della sicurezza contro la manipolazione della tecnologia blockchain.</i></p>
Intelligenza	<p><i>Argomenti</i></p> <ul style="list-style-type: none"> – Apprendimento con e senza supervisore.

<p>artificiale e machine learning</p>	<ul style="list-style-type: none"> – I modelli <i>white box</i>, <i>grey box</i> e <i>black box</i>. – I <i>task</i> di classificazione, regressione, predizione e <i>clustering</i>. – L'algoritmo <i>Least mean square</i> (LMS). – L'algoritmo <i>K-nearest neighbors</i> (KNN). – Le reti neurali artificiali (FNN) e la computazione neurale. – Le reti di convoluzione (CNN) e i meccanismi generativi. <p><i>Obiettivi</i> Conoscere i concetti che stanno alla base dell'intelligenza artificiale e del <i>machine learning</i>; l'apprendimento automatico del <i>machine learning</i>: saper individuare potenzialità e limiti di questi strumenti; saper replicare esempi di <i>training</i> di reti neurali artificiali per la risoluzione di <i>task</i> di classificazione o regressione; saper costruire piccole applicazioni, come ad esempio un programma di riconoscimento facciale, facendo capo a librerie specifiche per il <i>machine learning</i>. Sviluppare un senso critico rispetto al rapporto uomo-macchina, ragionando ad esempio sull'oggettivazione della realtà attraverso gli algoritmi o sull'analogia tra reti neurali e cervello organico.</p>
<p>Multimedia</p>	<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>. <p><i>Obiettivi</i> Riconoscere e saper trattare i diversi formati grafici, audio e video. Riconoscere le implicazioni sociali e giuridiche dell'uso di contenuti multimediali.</p>
<p>Programmazione: metodi matematici</p>	<p><i>Argomenti</i></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. <p><i>Obiettivi</i> Essere in grado di implementare correttamente algoritmi numerici e saperne valutare la correttezza e il costo computazionale.</p>

13.5. Valutazione

Nel corso del biennio l'allievo deve dimostrarsi capace di recepire, riprodurre e sviluppare autonomamente i principali aspetti dei temi trattati. Il raggiungimento di questi obiettivi è verificato attraverso:

- esercitazioni in classe e compiti da svolgere a casa;
- la presentazione di lavori di approfondimento personale o a gruppi;
- la partecipazione attiva dell'allievo alle lezioni;
- lavori scritti e lavori al computer.