

Curricolo d'Istituto – Liceo Scientifico "A. Oriani"

INFORMATICA – PRIMO BIENNIO – Indirizzo Scientifico Scienze applicate

COMPETENZE	ABILITA'	CONTENUTI
<p>PRIMO BIENNIO Nel primo biennio sono usati gli strumenti di lavoro più comuni del computer insieme ai concetti di base ad essi connessi. Lo studente è introdotto alle caratteristiche architettoniche di un computer: i concetti di hardware e software, gli elementi funzionali della macchina di Von Neumann: CPU, memoria, dischi, bus e le principali periferiche. Conosce il concetto di sistema operativo, le sue funzionalità di base e le caratteristiche dei sistemi operativi più comuni; il concetto di processo come programma in esecuzione, il meccanismo base della gestione della memoria. Lo studente conosce gli elementi costitutivi di un documento elettronico e i principali strumenti di produzione. Apprende la struttura e i servizi di Internet. Lo studente è introdotto ai principi alla base dei linguaggi di programmazione e gli sono illustrate le principali tipologie di linguaggi e il concetto di algoritmo. Sviluppa la capacità di implementare un algoritmo in pseudocodice o in un particolare linguaggio di programmazione, di cui si introdurrà la sintassi.</p>	<p>CLASSE I Riconoscere le componenti fondamentali di un computer dal punto di vista logico-funzionale. Simulare l'esecuzione di istruzioni macchina da parte di CPU, RAM, bus dati e indirizzi. Utilizzare le funzionalità di Windows per la gestione dei file e delle cartelle e la personalizzazione di alcuni aspetti delle periferiche.</p> <p>Salvare i propri file utilizzando la rete della scuola. Acquisire una visione virtuale del PC come insieme di ambienti sempre più astratti e lontani dalla macchina fisica di base.</p> <p>Progettare semplici programmi che utilizzino le 3 strutture di sequenza, selezione, iterazione.</p>	<p>CLASSE I Riconoscere le componenti hardware fondamentali di un computer . <i>UA1. Il sistema di elaborazione</i> La macchina reale: hardware e software. Bit e byte. La macchina di Von Neumann: CPU, memoria centrale, bus dati, indirizzi e controllo, clock. Unità di I/O. Memorie ausiliarie: dischi magnetici. Il concetto di file. Le periferiche di input-output. Sistemi di numerazione binario ed esadecimale. I codici ASCII e Unicode. Il linguaggio macchina. Conoscere le principali tipologie di software. <i>UA2. Verso la macchina virtuale</i> Software di base e applicativo. Sistema operativo. Allocazione di file su disco. Gestione gerarchica delle memorie di massa. Il sistema operativo Windows. Pannello di controllo. Gestione risorse: gestire le cartelle e i file. Il cestino. Creazione di collegamenti. Gli ambienti oltre il sistema operativo: linguaggi di programmazione e strumenti di produttività individuale. <i>UA3 Documenti elettronici e strumenti di produzione</i> Word processing: tabelle, indici. Il concetto di ipertesto, la navigazione nei documenti. Foglio elettronico: Riferimenti assoluti e relativi. Funzioni di base e logiche. Grafici: istogrammi, torta, linee e a dispersione. Presentazioni: struttura diapositive, animazioni e transizioni. Conosce gli strumenti per la rappresentazione degli algoritmi. <i>UA4. Dal problema all'algoritmo</i> Problema, risolutore, esecutore, utente. Caratteristiche di un algoritmo. Dati e istruzioni. Rappresentazione degli algoritmi. Sintassi e semantica dei linguaggi artificiali. Teorema di Bohn-Jacopini: strutture di sequenza, selezione, iterazione e loro combinazioni.</p>
	<p>CLASSE II Comprendere i passi logici necessari alla risoluzione di un problema, scomponendolo in sottoproblemi. Sapere scrivere funzioni, identificando parametri di ingresso e di uscita rispetto al loro utilizzo. Riconoscere sintassi e semantica delle istruzioni di base del linguaggio scelto. Utilizzare in modo consapevole le funzionalità di editing, compile e running dell'ambiente di programmazione scelto. Usare gli strumenti di debug per verificare la correttezza di una soluzione.</p> <p>Produrre semplici pagine HTML servendosi di un editor testuale. Leggere e modificare una pagina HTML esistente.</p>	<p>CLASSE II Conosce un linguaggio di programmazione di alto livello. <i>UA1. Dall'algoritmo al programma C</i> Le fasi di realizzazione di un programma: editor, programma sorgente, compilatore/interprete, programma oggetto. Linker, eseguibile, loader. Dichiarazione di variabili, tipi di dato, istruzioni di I/O, assegnazione. Strutture if, switch, for, while, do-while. Strumenti di debug. Gestione di sequenze di dati con o senza terminazione. Scomposizione funzionale, metodologia top-down, riuso del software, valore di ritorno, parametri attuali e formali: procedure e funzioni. Passaggio di parametri per indirizzo e per valore. <i>UA2 – Vettori e algoritmi notevoli</i> Definizione e parametrizzazione di vettori, gestione della terminazione. Algoritmi notevoli: scansione, media, massimo e minimo, array speculari, ricerca sequenziale e binaria, metodi di ordinamento (insertion, selection e bubble sort), shift a destra/sinistra di elementi, fusione di vettori. <i>UA3 – Stringhe e algoritmi notevoli</i> Gestione delle stringhe. Algoritmi di gestione delle stringhe in sola lettura, modifica della stringa senza e con modifica della lunghezza. Conosce la struttura dell'HTML. <i>UA3 - Internet ed Elementi di base di HTML</i> Architettura client/server. Il web. Protocollo HTTP; il concetto di URL. Il concetto di ipertesto. Linguaggio HTML: marcatori ed attributi., inserimento di testo, immagini e link. Uso degli stili (CSS). Cenni al linguaggio javascript (form) e realizzazione di programmi attinenti alle materie scientifiche.</p>

Curricolo d'Istituto – Liceo Scientifico "A. Oriani"

INFORMATICA – SECONDO BIENNIO – Indirizzo Scientifico Scienze applicate

COMPETENZE	ABILITA'	CONTENUTI
<p>Per il secondo biennio le strutture dati (vettore, matrice e record) sono presentate in relazione a semplici problemi da risolvere.</p> <p>Sono affrontati gli algoritmi notevoli per la gestione dei vettori, senza uso di puntatori, con lo scopo di modellizzare problemi la cui soluzione richiede dati strutturati. I file di record vengono sviluppati come termine di confronto con i database.</p> <p>Lo studente è in grado di gestire semplici applicazioni con interfaccia visuale che rispondono ad eventi e che implementano/utilizzano classi di oggetti (il linguaggio può essere quello del primo biennio). E' inoltre in grado di modellare i dati presenti in un determinato contesto, progettando a livelli diversi di astrazione: livello concettuale, più vicino alle esigenze del committente, livello logico, che implica la scelta di un legame di tipo matematico tra i dati con le relative operazioni, livello fisico, che riguarda il modo in cui i dati vengono memorizzati sui supporti. Sa infine utilizzare un linguaggio non procedurale per l'implementazione, la</p>	<p>CLASSE III Scrivere funzioni di manipolazione sequenziale di vettori e stringhe. Scrivere algoritmi di manipolazione di più vettori con indici differenti. Implementare e gestire dati strutturati a matrice. Implementare e gestire dati strutturati come vettori di record. Utilizzare le funzionalità di base per i file sequenziali e ad accesso diretto.</p> <p>Organizzare correttamente file e risorse degli ipertesti separando forma e contenuto. Creare algoritmi in JavaScript.</p>	<p>CLASSE III Conosce in modo approfondito un linguaggio di alto livello, la struttura di un record e dei file e li sa gestire. <i>UA1- Record, vettori di record, file di record, puntatori, memoria dinamica</i> Gestione di array a due indici: generazione, scansione, ricerca, verifica simmetria, media righe/colonne, quadrato magico, matrici triangolari. Definizione ed uso dei record. Definizione di vettori di record. Algoritmi su vettori di record (ordinamento, ricerca,...). Gestione di file di record: apertura e chiusura di un file, gestione della marca di EOF. Lettura e scrittura di file di record. Ricerca sequenziale su file. Accesso diretto al record. Cancellare record logicamente e fisicamente. File di testo. Gestione dinamica della memoria. Creare fogli di stile CSS e creare algoritmi in JavaScript. <i>UA2- HTML/CSS/JAVASCRIPT</i> I fogli di stile (CSS): in linea, incorporati, collegati Java Script: algoritmi embedded</p> <p>Robotica UA3 - ARDUINO Ambiente di simulazione : tinkercad. Ambiente di programmazione: arduino IDE. Alcuni sensori e attuatori: led, buzzer, sensore di luce, pulsante, display LCD, servomotore.</p>
<p>La competenza descritta nella prima colonna è in grado di essere raggiunta dallo studente in grado di:</p>	<p>CLASSE IV Gestire una soluzione software in ambiente console. Scrivere funzioni di risposta agli eventi utilizzando gli strumenti della programmazione strutturata. Implementare una classe con i relativi metodi e proprietà.</p> <p>Rilevare i limiti di una gestione non integrata degli archivi. Progettare semplici modelli E-R con alcune entità e le relative associazioni. Riconoscere i tipi di associazioni tra entità. Tradurre un modello E-R in un modello logico relazionale. Definire il modello fisico.</p>	<p>CLASSE IV Conosce un linguaggio di programmazione a oggetti. <i>UA1 - Le basi della programmazione a oggetti</i> Classi di oggetti: metodi e proprietà. Classe, attributi, metodi Incapsulamento, Information Hiding, Ereditarietà, Polimorfismo. Costruttore e distruttore. Template e STL</p> <p>Conosce un linguaggio per la gestione di un database. <i>UA2 - Sistemi informativi e modelli di dati</i> Ciclo di vita di un sistema informativo. Dati e informazioni, schema ed istanza dei dati, terminologia specifica sulle basi di dati. Caratteristiche di un DBMS. Definizione di modello concettuale, modello logico, modello fisico. Viste. Indipendenza tra i livelli. Linguaggi per database: DDL, DML, Query Language. Modello concettuale: il modello E-R, entità, associazioni, attributi, chiavi. Associazioni binarie tra entità di tipo 1-1,1-N,N-M,ISA; associazioni n-arie. Associazioni ricorsive. Rappresentazione grafica del modello concettuale attraverso gli schemi E-R. Modello logico: il modello relazionale, terminologia e rappresentazione tabellare. Regole di derivazione del modello relazionale dal modello E-R. Vincoli di integrità referenziale. Operazioni relazionali: selezione, proiezione, join. Il problema delle anomalie. Il processo di normalizzazione fino alla terza forma normale. Modello fisico: costruzione del modello fisico. Tipi di dati e dimensionamento degli attributi. Scelta delle chiavi, indicizzazione. <i>UA3 - SQL</i> Creazione di tabelle e aggiornamento dei dati. Data definition language (DDL): comandi CREATE, ALTER, DROP. Data manipulation language (DML): comandi INSERT, UPDATE, DELETE. Interrogazione dei dati: il comando SELECT, operazioni di selezione, proiezione, join. Operatori AND, OR, NOT, LIKE, BETWEEN. Funzioni di aggregazione: COUNT, SUM, AVG, MAX, MIN. ORDER BY. Raggruppamenti con GROUP BY, clausola HAVING</p>

<p>manipolazione e l'interrogazione di schemi relazionali di basi di dati (SQL).</p>	<p>Usare gli operatori fondamentali dell'algebra relazionale (selezione, proiezione, join). Riconoscere basi di dati normalizzate/non normalizzate. Codificare e validare in linguaggio SQL le operazioni di selezione, proiezione, join. Codificare e validare in linguaggio SQL raggruppamenti, ordinamenti, funzioni di aggregazione. Codificare e validare in linguaggio SQL condizioni sui raggruppamenti. Utilizzare un particolare DBMS per creare tabelle e impostare interrogazioni SQL.</p>	<p><i>UA4 – Presentazione di un DBMS specifico: Access, MySql, Postgres,...</i> Definizione di tabelle, attributi, vincoli sugli attributi, associazioni tra tabelle, vincoli di integrità referenziale, indici, gestione di query, aggiornamenti.</p>
--	---	--

Curricolo d'Istituto – Liceo Scientifico "A. Oriani"
INFORMATICA – CLASSE V – Indirizzo Scientifico Scienze applicate

COMPETENZE	ABILITA'	CONTENUTI
<p>Al termine del percorso liceale lo studente padroneggia i più comuni strumenti software per il calcolo, la ricerca e la comunicazione in rete, la comunicazione multimediale, l'acquisizione e l'organizzazione dei dati, applicandoli in una vasta gamma di situazioni, ma soprattutto nell'indagine scientifica, e scegliendo di volta in volta lo strumento più adatto. Ha una sufficiente padronanza di uno o più linguaggi per sviluppare applicazioni semplici, ma significative, di calcolo in ambito scientifico. Comprende la struttura logico/funzionale della struttura fisica e del software di un computer e di reti locali, tale da consentirgli la scelta dei componenti più adatti alle diverse situazioni e le loro configurazioni, la valutazione delle prestazioni, il mantenimento dell'efficienza.</p>	<p>Utilizzo dei principali algoritmi del calcolo numerico e dei principi teorici della computazione.</p> <p>Sviluppo di semplici simulazioni come supporto alla ricerca scientifica (studio quantitativo di una teoria, confronto di un modello con i dati...).</p> <p>Riconoscere e gestire le tematiche relative alle reti di computer, ai protocolli di rete, alla struttura di internet e dei servizi di rete.</p>	<p>Algoritmi del calcolo numerico. Applicazioni di simulazione. <i>UA1 - Calcolo numerico</i> Esercitazioni di analisi numerica tipo: interpolazione; binomio di Newton; calcolo costanti (n, e); calcolo di integrali (metodo dei trapezi, delle parabole, Montecarlo); algoritmo di bisezione, secanti, tangenti, punto unito Complessità computazionale: confronto tra algoritmi di analisi numerica o di ordinamento o di ricerca Simulazione:</p> <ul style="list-style-type: none"> • raccolta dati e costruzione del relativo database per scopi di ricerca scientifica (fisica, geologica, esempio: costruzione di un db sulle rocce) • costruzione di un'applicazione grafica per simulare le montagne russe • elaborazione di dati strutturati a matrice con applicativi (tipo Matab) <p>Reti di computer. <i>UA2 - Le architetture di rete</i> Caratteristiche delle reti locali; topologie di rete. Mezzi fisici di trasmissione: cavo coassiale, doppino, fibre ottiche, etere. Apparati di rete: schede di rete, hub, switch, router. Il modello client/server. Il modello ISO/OSI a strati, con particolare riferimento al TCP/IP. Mac address. Il protocollo IP: indirizzi IP, subnet mask, reti private, sottoreti. Classi di indirizzi IP, reti private. Configurazione degli indirizzi IP in una rete locale. Protocolli del livello Internet: protocollo ARP e ICMP. Protocolli del livello di trasporto: protocolli TCP e UDP <i>UA3 - Applicazioni di rete</i> DNS e instradamento dei pacchetti. I servizi del livello di applicazione: cenni al trasferimento dei file. La navigazione nel web: HTTP, URL, HTML e browser. La gestione della posta elettronica: protocolli SMTP, POP3, IMAP. Teoria e applicazione della sicurezza nel trasferimento dati: cenni a sistemi di sicurezza e crittografia, algoritmo RSA e sistemi a chiave pubblica/privata, certificati digitali, protocolli per la sicurezza. Dal database locale a quello in rete: accesso via Web ad una base di dati su server. Funzionamento di una applicazione lato server: architettura a tre livelli. Implementazioni di pagine dinamiche lato server.</p>