

Materia	INFORMATICA		
Classe	3A LATTANZIO	Anno scolastico	2018-19
Libro di testo			
Autore	LORENZI AGOSTINO / MORIGGIA VITTORIO / RIZZI ANDREA		
Titolo	INFORMATICA PER ISTITUTI TECNICI TECNOLOGICI vol, A	ISBN	9788826818382
Casa Editrice	ATLAS	Prezzo	19,80

PROGRAMMAZIONE

DOCENTI	Insegnante teorico	PAOLA PASCUCCI	
	Insegnante tecnico pratico	FRANCESCO SALIS	

RISULTATI DI APPRENDIMENTO ATTESI:

In riferimento alle linee guida si elencano i risultati di apprendimento individuati per la materia:

Competenze:

- Sviluppare applicazioni informatiche in modalità console con uno specifico linguaggio di programmazione
- Organizzare opportunamente i dati da elaborare
- Documentare i passaggi fondamentali delle applicazioni sviluppate
- Utilizzare a livello base la lingua inglese per la consultazione di manuali tecnici del settore
- Utilizzare le risorse in rete per la ricerca autonoma di soluzioni ad eventuali problemi applicativi
- Collaborare al raggiungimento degli obiettivi all'interno del gruppo partecipando all'analisi e alla realizzazione delle soluzioni dei problemi proposti

Conoscenze:

- Relazioni fondamentali tra macchine, problemi, informazioni e linguaggi
- Linguaggi e macchine a vari livelli di astrazione
- Paradigmi di programmazione
- Logica iterativa e ricorsiva
- Principali strutture dati e loro implementazione
- Strumenti per lo sviluppo del software e supporti per la robustezza dei programmi
- Lessico e terminologia tecnica di settore anche in lingua inglese

Abilità:

- Progettare e implementare algoritmi utilizzando diverse strutture di dati
- Analizzare e confrontare algoritmi diversi per la soluzione dello stesso problema
- Scegliere il tipo di organizzazione dei dati più adatto a gestire le informazioni in una situazione data
- Utilizzare il lessico e la terminologia tecnica di settore anche in lingua inglese
- saper comunicare e condividere le proprie modalità di ragionamento
- saper assumere la responsabilità di prendere decisioni e operare scelte

ARTICOLAZIONE ORARIA

Sono previste **3** ore di teoria e **3** ore di esercitazioni in laboratorio

ANALISI DELLA SITUAZIONE DI PARTENZA DELLA CLASSE

Profilo generale della classe

La classe è composta da 24 alunni, tutti provenienti dallo stesso istituto ma da due classi seconde diverse. Inoltre sono presenti 4 alunni ripetenti e un alunno non ha mai frequentato. Quindi la classe è effettivamente composta da 23 alunni.

Alunni con disturbi specifici dell'apprendimento:

E' presente un alunno con DSA.

Livelli di partenza rilevati

Da una prima valutazione si evidenziano molti alunni con buone capacità che seguono con interesse e impegno. Pochissimi mostrano poco interesse, in particolare nello svolgere i compiti assegnati per casa. Si evidenziano due situazioni molto particolari di due alunni che pur applicandosi a seguendo con attenzione le lezioni, potrebbero non avere sufficienti capacità logiche e conoscenze pregresse per affrontare con risultati sufficienti lo studio di questa materia.

UNITÀ DISCIPLINARI DEL PERCORSO FORMATIVO

Le unità didattiche prevedono lezioni teoriche e lezioni pratiche di laboratorio

1. Unità didattica : Introduzione alla materia		COMPETENZE • Organizzare opportunamente i dati da elaborare
Periodo	Conoscenze	Abilità
Settembre Ottobre	<p>1.1. Informatica e informazioni Dati e informazioni la codifica delle informazioni l'informatica e i suoi ambiti</p> <p>1.2. Affrontare i problemi: il problem solving Che cosa è un problema? capire il problema elaborare un piano per la risoluzione eseguire il piano programmato</p>	<ul style="list-style-type: none"> • Conoscere e utilizzare consapevolmente i termini tecnici di base • saper interpretare i dati • saper codificare informazioni • conoscere gli ambiti di applicazione dell'informatica • saper riformulare il testo di un problema • individuare dati, condizioni, incognite e obiettivo • stabilire relazioni/nessi tra fatti, dati, termini • formulare ipotesi e strategie risolutive • saper eseguire le operazioni necessarie per passare dai dati ai risultati • verificare il lavoro fatto
2. Unità didattica : Gli algoritmi		COMPETENZE • Individuare le azioni da compiere • Documentare i passaggi fondamentali delle applicazioni sviluppate
Periodo	Conoscenze	Abilità
Settembre Ottobre Novembre	<p>2.1. Gli algoritmi Dal problema all'algoritmo algoritmi ed esecutori la rappresentazione degli algoritmi schemi di composizione delle operazioni di un algoritmo schemi delle strutture di controllo dell'algoritmo la macchina di Turing come esecutore di algoritmi cenni sulla valutazione della complessità degli algoritmi</p>	<ul style="list-style-type: none"> • Individuare i dati di input e l'output richiesto • conoscere le regole di scrittura della pseudocodifica e i simboli del DaB • individuare il ruolo di risolutore, esecutore, utilizzatore • distinguere operazioni elementari e complesse in funzione dell'esecutore • saper utilizzare gli schemi DaB per la rappresentazione delle strutture di controllo dell'algoritmo • saper rappresentare un algoritmo in DaB e in pseudocodice • saper riconoscere la "macchina" di Turing come esecutore di algoritmi • conoscere i criteri di massima per la valutazione della complessità di un algoritmo

3. Unità didattica : Linguaggi di programmazione		COMPETENZE • Evoluzione dei linguaggi di programmazione
Periodo	Conoscenze	Abilità
Ottobre Novembre	<p>3.1. Linguaggi naturali e formali linguaggi di programmazione evoluzione dei linguaggi di programmazione paradigmi di programmazione</p> <p>3.2. fasi di sviluppo di un programma codice sorgente e codice eseguibile compilatori ed interpreti</p>	<ul style="list-style-type: none"> • Conoscere gli aspetti sintattici e semantici dei linguaggi • saper illustrare le caratteristiche dei linguaggi di programmazione • saper individuare le fasi di edit, compile, run • riconoscere il file sorgente ed eseguibile • conoscere le modalità di funzionamento di un compilatore e di un interprete

4. Unità didattica : Il linguaggio di programmazione C++		COMPETENZE • Sviluppare semplici applicazioni informatiche in modalità console con uno specifico linguaggio di programmazione
Periodo	Conoscenze	Abilità
Novembre Dicembre Gennaio	<p>4.1. Caratteristiche del linguaggio Struttura fondamentale di un programma ambiente di sviluppo variabili e costanti espressioni e condizioni operatori ed espressioni logiche; tabella di verità operazioni standard di input e output controllo del flusso di esecuzione: le istruzioni che implementano le strutture di controllo dell'algoritmo sequenza,selezione, ripetizione funzioni della libreria matematica</p> <p>4.2. Gli array in C++ Array mono dimensionali; indici vettore: definizione, memorizzazione, elaborazione stringhe di caratteri in stile C</p> <p>4.3. Algoritmi di ordinamento e ricerca Ordinamento: exchange-sort e bubble-sort complessità degli algoritmi di ordinamento ricerca: completa e binaria</p>	<ul style="list-style-type: none"> • Saper utilizzare l'ambiente di sviluppo per programmi C++ • saper codificare i tipi di dato elementari, gli operatori aritmetici, logici e di relazione • saper utilizzare le istruzioni standard di input e output • saper codificare le strutture di controllo dell'algoritmo • saper individuare la differenza tra le istruzioni iterative • saper codificare, compilare ed eseguire un programma • saper capire e correggere gli errori di compilazione • saper costruire i piani di prova per il test del programma • saper utilizzare vettori nella realizzazione di algoritmi • saper valutare l'efficienza degli algoritmi di ordinamento e di ricerca

5. Unità didattica : Il linguaggio di programmazione C++: concetti avanzati		COMPETENZE • Sviluppare applicazioni informatiche in modalità console con uno specifico linguaggio di programmazione scomponendo il problema in sottoproblemi
Periodo	Conoscenze	Abilità
Gennaio Febbraio Marzo Aprile Maggio	<p>5.1 Le funzioni in C++ Definizione e invocazione di una funzione passaggio dei parametri per valore e per riferimento prototipazione delle funzioni overloading dei nomi delle funzioni</p>	<ul style="list-style-type: none"> • saper scomporre un problema in funzioni • saper strutturare e codificare un programma con funzioni • saper riutilizzare più volte la stessa funzione con parametri diversi • saper utilizzare vettori e matrici nella realizzazione di algoritmi • saper definire e utilizzare il tipo di dato

	<p>5.2 Gli array in C++ Array bidimensionali; indici matrici: definizione, memorizzazione, elaborazione matrici quadrate; diagonale principale e secondaria array come parametri di funzioni stringhe di caratteri in stile C</p> <p>5.3 Le strutture in C++ Le strutture come tipi di dato definiti dall'utente tabelle come array di strutture</p> <p>5.4 La ricorsività (cenni) Induzione e ricorsione funzioni ricorsive ricorsione e array</p> <p>5.5 I file Organizzazione e modalità di accesso gestione dei file in C++ tecniche per la gestione dei file di testo</p>	<p>struttura</p> <ul style="list-style-type: none"> • saper scegliere i tipi di dato rispondenti alla realtà da gestire • conoscere la differenza tra un algoritmo iterativo e ricorsivo • saper trasformare un algoritmo ricorsivo in iterativo • la persistenza dei dati in memoria di massa • saper individuare l'organizzazione di un file e il metodo di accesso • saper realizzare programmi in C++ che utilizzano file
--	--	---

6. Unità didattica : Sviluppo web	COMPETENZE	
	<ul style="list-style-type: none"> • Sviluppare semplici applicazioni a interfaccia grafica 	
Periodo	Conoscenze	Abilità
Marzo Aprile Maggio	<p>6.1 Concetti generali di sviluppo web Browser, motore di ricerca, WWW, URL, Web server, regole generali di comunicazione sul web importanza delle tecniche SEO <i>meta tag</i> e <i>tag header</i> in ottica SEO ottimizzazione immagini lato SEO</p> <p>6.2 Il linguaggio HTML evoluzione del linguaggio HTML standard W3C la struttura della pagina HTML testo, elenchi e tabelle, link le immagini tag semantici</p> <p>6.3 CSS Utilità dei fogli di stile inserimento e interpretazione di codice CSS selettori <i>id</i> e <i>class</i> gestire le proprietà del testo gestire lo stile dei link sfondi e immagini il modello a scatola</p>	<ol style="list-style-type: none"> 1. Saper distinguere lato client e server in un'applicazione web 2. individuare i software utilizzati come browser e motori di ricerca 3. formazione e utilizzo dell'url 4. conoscere le regole basilari per la comunicazione efficace nel web 5. comprendere l'importanza e conoscere le principali tecniche SEO 6. saper progettare il layout di pagine web 7. saper utilizzare correttamente i tag HTML 5 per strutturare una pagina web con testo, immagini, elenchi, tabelle, link e sfondi 8. saper utilizzare codice CSS 3 interno ed esterno alla pagina HTML <ul style="list-style-type: none"> • saper utilizzare lo strumento di validazione del W3C

OBIETTIVI MINIMI (comuni a tutte le classi parallele, individuati dal Dipartimento)

Si ritiene fondamentale il conseguimento degli obiettivi minimi per il passaggio alla classe successiva

CONOSCENZE	ABILITA'
<ul style="list-style-type: none">• Dati e informazioni• Concetti di problem solving• Algoritmi ed esecutori• La rappresentazione degli algoritmi• Schemi delle strutture di controllo dell'algoritmo• Codifica dell'algoritmo• Fasi di sviluppo di un programma• Ambiente di sviluppo di un programma in linguaggio C++• Operazioni standard di input e output• Codifica delle strutture di controllo dell'algoritmo (sequenza, selezione, ripetizione)• Le funzioni in C++ : definizione e chiamata• Passaggio dei parametri per valore e per riferimento• Dati elementari e strutturati: gestione di stringhe di caratteri e di array monodimensionali• Algoritmi standard di ricerca e ordinamento• Il Web: browser, motori di ricerca, URL e DNS• Struttura di una pagina HTML• Tag per inserimento di testo, elenchi, tabelle, link e immagini	<ul style="list-style-type: none">• Collaborare attivamente in un gruppo di lavoro• Riconoscere e usare correttamente la terminologia di base dell'Informatica• Formulare i passi per risolvere problemi, individuando i dati necessari e l'obiettivo da raggiungere• Impostare problemi con tecnica procedurale adeguata all'esecutore automatico PC• Rappresentare l'algoritmo risolutivo di un problema con diagrammi di flusso• Verificare la correttezza di una soluzione • Interagire con il Sistema Operativo installato sul PC in uso• Utilizzare l'ambiente di programmazione• Codificare algoritmi in C++ utilizzando le strutture di controllo dell'algoritmo, le funzioni, le variabili semplici, le stringhe e gli array unidimensionali• Documentare software a livello elementare• Navigazione consapevole in Internet per il recupero di informazioni• Distinguere lato client e server in un'applicazione web• Strutturare una pagina web statica con testo, immagini, elenchi, tabelle, link e sfondi

FORMATI DIDATTICI PREVISTI

- Lezione frontale e interattiva con utilizzo della LIM.
- Tecnica di problem solving per mettere gli studenti di fronte a situazioni aperte che vengono affrontate per tentativi anche empirici.
- Gruppi di lavoro
- Attività pratica di laboratorio in cui trovano adeguata verifica le metodologie sopra elencate

Le lezioni si svolgeranno in aula e in laboratorio di informatica come previsto dall'orario scolastico.

STRUMENTI E MATERIALI DIDATTICI

- Libro di testo
- schede di lavoro proposte dai docenti
- eventuali dispense di approfondimento e/o completamento degli argomenti proposti nonché esercizi da svolgere potranno essere depositate nella "AREA DOWNLOAD - Lezioni e Appunti " del sito dell'Istituto
- Internet per la ricerca autonoma di soluzioni a quesiti proposti

TIPOLOGIA DELLE PROVE DI VERIFICA PREVISTE

Le prove di verifica che si intende utilizzare sono:

- Prove pratiche (laboratorio): lavori individuali e/o di gruppo, relazioni individuali.
- Prove scritte di diverso tipo:
 1. quesiti a risposta singola. Le risposte devono essere autonomamente formulate dallo studente;
 2. quesiti con risposta a scelta multipla. L'alunno deve scegliere tra più opzioni la risposta giusta
 3. realizzazione di algoritmi, rappresentazione con diagramma di flusso o pseudocodifica, codifica in linguaggio di programmazione.
- Prove orali: interrogazioni, interventi significativi e partecipazione al dialogo educativo.

Numero **minimo** di prove per quadrimestre: **4** (1 pratica, 2 scritte, 1 orale).

CRITERI DI VALUTAZIONE, INDICATORI E GRIGLIE

Per i criteri di valutazione ci si atterrà a quelli illustrati nel P.T.O.F.

Gli indicatori di riferimento sono:

per la produzione scritta

- | | |
|--|-----------------------------|
| a) correttezza e proprietà di linguaggio | d) coerenza logica |
| b) pertinenza alla traccia | e) rielaborazione personale |
| c) conoscenza dei contenuti | |

per il colloquio:

- | | |
|--|---|
| a) correttezza nell'uso dei linguaggi specifici | c) capacità di effettuare collegamenti |
| b) conoscenza degli argomenti e capacità di individuarne gli elementi fondanti | d) capacità di esprimere giudizi motivati |

per l'attività di laboratorio:

- | | |
|-----------------|----------------------------|
| a) cooperazione | c) responsabilità |
| b) autonomia | d) rispetto delle consegne |

Per le prove scritte e le prove pratiche di laboratorio si seguiranno le seguenti regole:

- ogni quesito sarà accompagnato da uno specifico punteggio stabilito dal docente per i casi di quesito corretto/errato/assente
- I punti totali ottenuti dallo studente saranno tradotti in punti decimi secondo una scala pubblicata dal docente contestualmente alla prova

Per lo scritto che richiede la produzione di un programma, verranno presi in considerazione i seguenti elementi (con relativa incidenza sul voto finale):

- | | |
|---|-----|
| • corretta e completa individuazione dell'I/O | 20% |
| • corretta individuazione e realizzazione dell'algoritmo risolutore | 60% |
| • documentazione, indentazione ed utilizzo di commenti | 15% |
| • originalità della soluzione e/o ottimizzazioni introdotte | 5% |

Laddove non presente uno o più degli elementi valutativi precedenti, la relativa percentuale verrà ripartita sulle altre componenti.

Per le interrogazioni orali verranno presi in considerazione i seguenti elementi (con relativa incidenza sul voto finale):

- | | |
|---|-----|
| • padronanza del linguaggio tecnico | 30% |
| • conoscenza degli argomenti (livello di approfondimento e capacità di cogliere l'essenza del tema esposto) | 50% |
| • capacità di orientamento autonomo, cogliere nessi e operare raccordi | 20% |

I voti saranno espressi in decimi secondo l'intervallo approvato dal Collegio dei Docenti

ATTIVITÀ DI RECUPERO, SOSTEGNO, APPROFONDIMENTO

In itinere saranno svolte attività di recupero a seconda delle necessità. Durante le settimane di recupero deliberate del Collegio docenti e definite dal Consiglio di Classe, è prevista la diversificazione dell'attività didattica, nelle ore di compresenza, con interventi di sostegno per gli studenti in difficoltà e di approfondimento per gli altri. Considerando che le principali difficoltà riscontrate fanno riferimento alla comprensione del testo e all'individuazione della strategia risolutiva dei problemi, si proporrà un'attività di "laboratorio di problem solving".

ATTIVITÀ DI ALTERNANZA SCUOLA LAVORO

La programmazione disciplinare potrebbe subire delle modifiche in base alle attività che la classe dovrà svolgere per l'Alternanza Scuola Lavoro. Si aderirà alle iniziative proposte dalla commissione ASL e approvate dal Consiglio di Classe.

RAPPORTI CON LE FAMIGLIE

Gli incontri con le famiglie avverranno con il ricevimento in orario antimeridiano e i 2 incontri generali pomeridiani stabiliti in sede di programmazione delle attività scolastiche.