

## PIANO DI MATERIA

Materia	INFORMATICA		
Classe	3D LATTANZIO	Anno scolastico	2018-19
Libro di testo			
Autore	A. Lorenzi – V. Moriggia		
Titolo	Informatica per Istituti Tecnici Tecnologici vol.A	ISBN	978-88-268-1838-2
Casa Editrice	Atlas	Prezzo	18,80

### 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
- Principali strutture dati e loro implementazione
- Strumenti per lo sviluppo del software
- 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
- Implementazione di algoritmi in linguaggio C++
- Utilizzare il lessico e la terminologia tecnica di settore anche in lingua inglese

### ARTICOLAZIONE ORARIA

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

### SITUAZIONE INIZIALE

#### Rilevazione dei livelli di partenza

Tipologia di prova utilizzata per rilevare i livelli di partenza: prova strutturata con quesiti relativa ai contenuti della materia *Tecnologie informatiche* del primo anno e quesiti di logica.

#### Eventuali percorsi interdisciplinari

All'interno dei singoli CdC si verificherà l'opportunità di proporre lavori pluridisciplinari.

### UNITÀ DISCIPLINARI DEL PERCORSO FORMATIVO

La dinamicità della tecnologia informatica suggerisce di privilegiare quanto è meno soggetto all'usura del tempo.

Percorso formativo

Conoscenze	Abilità
<b>Unità didattica 1: Introduzione alla materia</b>	
<p><b>1.Informatica e informazioni</b> Dati e informazioni la codifica delle informazioni l'informatica e i suoi ambiti</p> <p><b>2.Affrontare i problemi: il problem solving</b> Che cosa è un problema? capire il problema elaborare un piano per la risoluzione eseguire il piano programmato guardare indietro</p>	<p>3.Conoscere e utilizzare consapevolmente i termini tecnici di base 4.saper interpretare i dati 5.saper codificare informazioni 6.conoscere gli ambiti di applicazione dell'informatica 7.saper riformulare il testo di un problema 8.individuare dati, condizioni, incognite e obiettivo 9.stabilire relazioni/nessi tra fatti, dati, termini 10.formulare ipotesi e strategie risolutive 11.saper eseguire le operazioni necessarie per passare dai dati ai risultati 12.verificare il lavoro fatto 13.collaborare attivamente nel gruppo di lavoro 14.saper comunicare e condividere le proprie modalità di ragionamento 15.saper assumere la responsabilità di prendere decisioni e operare scelte</p>
<b>Unità didattica 2: Gli algoritmi</b>	
<p><b>Gli algoritmi</b> 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>	<p>1.Individuare i dati di input e l'output richiesto 2.conoscere le regole di scrittura della pseudocodifica e i simboli del DaB 3.individuare il ruolo di risolutore, esecutore, utilizzatore 4.distinguere operazioni elementari e complesse in funzione dell'esecutore 5.saper utilizzare gli schemi DaB per la rappresentazione delle strutture di controllo dell'algoritmo 6.saper rappresentare un algoritmo in DaB e in pseudocodice 7.saper riconoscere la "macchina" di Turing come esecutore di algoritmi 8.conoscere i criteri di massima per la valutazione della complessità di un algoritmo</p>
<b>Unità didattica 3: Linguaggi di programmazione</b>	
<p>Linguaggi naturali e formali linguaggi di programmazione evoluzione dei linguaggi di programmazione paradigmi di programmazione fasi di sviluppo di un programma codice sorgente e codice eseguibile compilatori ed interpreti</p>	<p>9.Conoscere gli aspetti sintattici e semantici dei linguaggi 10.saper illustrare le caratteristiche dei linguaggi di programmazione 11.saper individuare le fasi di edit, compile, run 12.riconoscere il file sorgente ed eseguibile 13.conoscere le modalità di funzionamento di un compilatore e di un interprete</p>
<b>Unità didattica 4: Il linguaggio di programmazione C++</b>	
<p><b>1.Caratteristiche del linguaggio</b> Struttura fondamentale di un programma ambiente di sviluppo variabili e costanti espressioni e condizioni</p>	<p>8.Saper utilizzare l'ambiente di sviluppo per programmi C++ 9.saper codificare i tipi di dato elementari, gli operatori aritmetici, logici e di relazione 10.saper utilizzare le istruzioni standard di input e</p>

<p>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  <b>2.Le funzioni in C++</b>                  Definizione e invocazione di una funzione                  passaggio dei parametri per valore e per riferimento                  prototipazione delle funzioni                  overloading dei nomi delle funzioni  <b>3.Gli array in C++</b>                  Array mono e bidimensionali; indici                  vettore: definizione, memorizzazione, elaborazione                  matrici: definizione, memorizzazione, elaborazione                  matrici quadrate; diagonale principale e secondaria                  array come parametri di funzioni                  stringhe di caratteri in stile C  <b>4.Le strutture in C++</b>                  Le strutture come tipi di dato definiti dall'utente                  tabelle come array di strutture  <b>5.Algoritmi di ordinamento e ricerca</b>                  Ordinamento: exchange-sort e bubble-sort                  complessità degli algoritmi di ordinamento                  ricerca: completa e binaria  <b>6.La ricorsività (cenni)</b>                  Induzione e ricorsione                  funzioni ricorsive                  ricorsione e array  <b>7.I file</b>                  Organizzazione e modalità di accesso                  gestione dei file in C++                  tecniche per la gestione dei file di testo</p>	<p>output                  11.saper codificare le strutture di controllo dell'algoritmo                  12.saper individuare la differenza tra le istruzioni iterative                  13.saper codificare, compilare ed eseguire un programma                  14.saper capire e correggere gli errori di compilazione                  15.saper costruire i piani di prova per il test del programma                  16.saper scomporre un problema in funzioni                  17.saper strutturare e codificare un programma con funzioni                  18.saper riutilizzare più volte la stessa funzione con parametri diversi                  19.saper utilizzare vettori e matrici nella realizzazione di algoritmi                  20.saper definire e utilizzare il tipo di dato struttura                  21.saper scegliere i tipi di dato rispondenti alla realtà da gestire                  22.saper valutare l'efficienza degli algoritmi di ordinamento e di ricerca                  23.conoscere la differenza tra un algoritmo iterativo e ricorsivo                  24.saper trasformare un algoritmo ricorsivo in iterativo                  25.la persistenza dei dati in memoria di massa                  26.saper individuare l'organizzazione di un file e il metodo di accesso                  27.saper realizzare programmi in C++ che utilizzano file</p>
<p><b>Unità didattica 5: Sviluppo web</b></p>	
<p><b>1.Concetti generali di sviluppo web</b>                  Browser, motore di ricerca, WWW, URL, Web server, Web spider/Web crawler                  regole generali di comunicazione sul web                  importanza delle tecniche SEO                  meta tag e tag header in ottica SEO                  ottimizzazione immagini lato SEO  <b>2.Il linguaggio HTML</b>                  evoluzione del linguaggio HTML                  standard W3C                  la struttura della pagina HTML                  testo, elenchi e tabelle, link                  le immagini                  tag semantici  <b>3.CSS</b>                  Utilità dei fogli di stile                  inserimento e interpretazione di codice CSS                  selettori id e class                  gestire le proprietà del testo                  gestire lo stile dei link                  sfondi e immagini</p>	<p>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                  9.saper utilizzare lo strumento di validazione del W3C</p>

il modello a scatola	
----------------------	--

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

**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
- Utilizzo della piattaforma e-learning dell'Istituto (in fase di attivazione) per la condivisione di materiali e la personalizzazione dei percorsi
- 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.O.F.

Gli indicatori di riferimento sono:

### per la produzione scritta

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

### per il colloquio:

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

### per l'attività di laboratorio:

- a) cooperazione
- b) autonomia
- c) responsabilità
- 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%  |

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<br>(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, 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"*.

### **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.

## ANALISI DELLA SITUAZIONE DI PARTENZA DELLA CLASSE

### Profilo generale della classe

La classe è composta da 25 alunni provenienti da due sezioni diverse ed un ripetente da un'altra scuola. L'atteggiamento scolastico è risultato fin da subito poco attento e partecipativo e poco rispettoso delle regole. Una buona parte degli studenti ha dimostrato di non avere la volontà di impegnarsi nello svolgimento dei compiti a casa oltre che di avere un metodo di studio non adeguato alle necessità della materia.

Si riscontra estrema difficoltà a coinvolgere la classe ad un dialogo attivo e costruttivo. Risultano poco efficaci gli sforzi rivolti allo svolgimento di lezioni che privilegiano lo svolgimento di esercizi in classe dando spunto ad osservazioni e confronti in modo da aiutare i ragazzi a trovare un metodo di studio più adeguato per la comprensione degli argomenti trattati.

### Livelli di partenza rilevati

I risultati delle prime valutazioni evidenziano pochi elementi di spicco, un diffuso livello di mediocrità ed anche diverse situazioni di estrema fragilità

## PIANIFICAZIONE DELLE ATTIVITÀ DIDATTICHE

Periodo	Argomento
Primo quadrimestre	U.D. 1, U.D. 2, U.D. 3, U.D. 4 (1,2,3), U.D. 5 (metà)
Secondo quadrimestre	U.D. 4 (4, 5, 6, 7) U.D. 5