MINISTERO DELLA ISTRUZIONE DELL'UNIVERSITA' E RICERCA
UFFICIO SCOLASTICO REGIONALE PER IL LAZIO
ISTITUTO DI ISTRUZIONE SUPERIORE "I.T.C. DI VITTORIO – I.T.I. LATTANZIO"
Via Teano, 223 - 00177 Roma ☎ 06121122405 / 06121122406- fax *062752492*
Cod. Min. RMIS00900E ✉ rmis00900e@istruzione.it - rmis00900e@pec.istruzione.it
Cod. fiscale 97200390587

PROGRAMMAZIONE DIDATTICA

Materia	**INFORMATICA**	
Classe	**3E Lattanzio**	
Anno scolastico	**2018/2019**	
Docenti	Insegnante teorico	***Pellecchia Carolina***
	Insegnante tecnico pratico	***Marras Franco***

OBIETTIVI SPECIFICI DI APPRENDIMENTO

Competenze:

- Acquisizione di una metodologia di sviluppo del problema
- Formalizzazione del procedimento risolutivo
- Validazione del procedimento risolutivo
- Individuazione delle risorse informatiche
- Utilizzo degli strumenti informatici in relazione all'analisi dei dati e alla modellazione dei problemi

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

- Saper implementare applicazioni in modalità console
- Applicare i principi di base della logica formale per studiare un algoritmo
- Codificare gli algoritmi e validare i programmi effettuando le necessarie correzioni
- Produrre un'efficace documentazione contestualmente allo sviluppo dei progetti software
- Identificare e ad applicare le metodologie e le tecniche per la progettazione e la realizzazione di pagine Web.

ARTICOLAZIONE ORARIA

Sono previste **3** ora 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 23 alunni di cui 1 proveniente da un'altra scuola, 1 ripetente proveniente da questo istituto.

Gli studenti provengono da due classi, 2E e 2N, di questo istituto. Pertanto il clima relazionale del gruppo classe è ancora non del tutto maturo. Dalle osservazioni svolte in questo periodo iniziale dell'anno scolastico, la classe si mostra interessata alla materia ma in generale non è costante nell'impegno. Questo atteggiamento, potrebbe compromettere i risultati.

Alunni con disturbi specifici dell'apprendimento:

In questa classe, sono presenti 3 alunni con bisogni educativi speciali.

UNITÀ DI APPRENDIMRENTO DEL PERCORSO FORMATIVO

Le unità di apprendimento prevedono lezioni teoriche e lezioni pratiche di laboratorio

<table>
<tr><td colspan="2">1. Unità di apprendimento: Cos'è l'informatica</td></tr>
<tr><td colspan="2">Competenze</td></tr>
<tr><td colspan="2">Avere una visione su:
sistema di elaborazione e logica di funzionamento,
caratteristiche di un problema e modo di approcciare alla soluzione,
funzioni complessive del sistema operativo.</td></tr>
<tr><td>Conoscenze</td><td>Abilità</td></tr>
<tr><td>Concetti di informazione, comunicazione, dato, elaborazione, sistema, modello, processo, processore
Principi generali per affrontare un problema
Pensiero computazionale
Il linguaggio delle cose
Caratteristiche e funzioni delle componenti fondamentali di un sistema di elaborazione.
Prompt dei comandi</td><td>Saper spiegare il significato dei termini fondamentali 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 di un problema
Saper individuare le unità che compongono un sistema di elaborazione
Inviare comandi al sistema operativo dal prompt dei comandi</td></tr>
<tr><td colspan="2">Periodo: settembre - ottobre</td></tr>
<tr><td colspan="2">2. Unità di apprendimento: Progettazione degli algoritmi</td></tr>
<tr><td colspan="2">Competenze</td></tr>
<tr><td colspan="2">Conoscere il concetto di algoritmo
Riconoscere le caratteristiche fondamentali delle istruzioni che compongono un algoritmo
Costruire algoritmi ben ordinati attraverso le strutture di controllo
Conoscere i diversi paradigmi di programmazione e gli aspetti evolutivi dei linguaggi di programmazione</td></tr>
<tr><td>Conoscenze</td><td>Abilità</td></tr>
<tr><td>Variabili e costanti, dati e azioni
La metodologia di lavoro nella formalizzazione dei problemi
Definizione e caratteristiche di algoritmo
Operazioni di input e di output
Gli operatori
Strumenti per la stesura di un algoritmo
L'individuazione dei dati di un problema
Le strutture di controllo</td><td>Saper distinguere all'interno di un problema tra variabili e costanti, tra dati e azioni
Utilizzare la pseudocodifica per rappresentare gli algoritmi
Rappresentare graficamente gli algoritmi con i diagrammi a blocchi
Costruire algoritmi strutturati
Rappresentare le strutture di controllo
Individuare le strutture di controllo più idonee per la soluzione di un problema</td></tr>
<tr><td colspan="2">Periodo: ottobre - novembre</td></tr>
<tr><td colspan="2">3. Unità di apprendimento: Linguaggio C++</td></tr>
<tr><td colspan="2">Competenze</td></tr>
<tr><td colspan="2">Individuare la struttura generale di un programma in linguaggio C++ e le caratteristiche principali dei dati, delle</td></tr>
</table>

istruzioni e degli operatori.
Scrivere i programmi in C++ utilizzando in modo corretto la sintassi delle istruzioni di input/output e delle strutture di controllo
Riconoscere le diverse fasi del lavoro di programmazione per codificare e validare gli algoritmi
Scomporre il programma in funzioni e riutilizzare più volte le stesse funzioni assegnando diversi valori ai parametri
Organizzare dati dello stesso tipo o di tipo diverso, associando ad ogni situazione problematica la struttura di dati più idonea

Conoscenze	Abilità

• **Struttura generale di un programma C++**
  • Tipi di dati numerici e non numerici
  • Dichiarazione delle costanti e delle variabili
  • Istruzione di assegnazione e operatori
  • Casting
  • Istruzioni di I/O
  • Fasi del lavoro di programmazione
  • Errori sintattici, lessicali, di run-time e logici
  • Importanza della documentazione
  • Codifiche delle tre strutture fondamentali: sequenza, alternativa, ripetizione
  • Strutture annidate di alternativa
  • Ripetizione precondizionale e ripetizione con contatore
  • Struttura di scelta multipla
• **Funzioni**
  • Funzioni con parametri
  • Passaggio di parametri per referenza e per valore
  • Dichiarazione dei prototipi di funzione
  • Definizione di risorse locali e globali
  • Regole di visibilità
  • Funzioni predefinite del linguaggio
  • Namespace e librerie di inclusione
  • Function overloading
  • Funzioni ricorsive
  • Enumerazioni
• **Array**
  • Array a due dimensioni
• Strutture
• Puntatori
• **Algoritmi di ordinamento e ricerca**
  • Ordinamento: exchange-sort e bubble-sort
  • complessità degli algoritmi di ordinamento
  • ricerca: completa e binaria
• **I file**
  • Organizzazione e modalità di accesso
  • gestione dei file in C++
  • tecniche per la gestione dei file di testo
	• Scrivere un programma C++ sintatticamente corretto
	• Scegliere il tipo di dato adatto a rappresentare le variabili
	• Validare un programma
	• Produrre programmi documentati
	• Individuare le strutture di controllo più idonee per la soluzione di un problema
	• Rappresentare la selezione
	• Annidare strutture di controllo
	• Esaminare un elenco di dati
	• Rappresentare le strutture derivate di ripetizione
	• Calcolare il valore massimo tra un insieme di valori
	• Utilizzare la struttura di scelta multipla
	• Sviluppare un programma introducendo le funzioni
	• Utilizzare il passaggio di parametri per referenza e per valore
	• Dichiarare le funzioni con i prototipi
	• Individuare le applicazioni pratiche delle regole di visibilità
	• Utilizzare le funzioni predefinite
	• Ridefinire una funzione con overloading
	• Individuare alcuni casi semplici di utilizzo di funzioni ricorsive
	• Creare un'enumerazione
	• Organizzare i dati in array a una o due dimensioni
	• Organizzare i dati in strutture
	• Utilizzare i puntatori

Periodo: novembre - marzo

4. Unità di apprendimento: Progettazione di pagine Web e fogli di stile

Competenze

Operare con informazioni, documenti e oggetti multimediali in formato Web da pubblicare nei siti Internet
Utilizzare strumenti e linguaggi per personalizzare il layout e lo stile delle pagine Web
Creare contenuti web in grado di adattarsi graficamente in modo automatico al dispositivo con il quale vengono visualizzati

Conoscenze	Abilità
• Concetti generali sulla programmazione Web	• Visualizzare una pagina Web in modalità offline

• Programmazione lato Client side
• Il ruolo del browser
• Standard W3C
• **Linguaggio HTML**
  • Tag del linguaggio
  • Struttura generale di una pagina HTML
  • Attributi dei tag
  • Formattazione del testo
  • Titoli, sottotitoli, paragrafi
  • Elenchi puntati e numerati
  • Tabelle/Div
  • Collegamenti ipertestuali
  • Immagini, audio, video
  • responsive web design (RWD)
• **Fogli di stile CSS**
  • Fogli di stile in linea, incorporati, collegati
  • Selettore, classe, identificatore
• Accessibilità e usabilità
	• Visualizzare il codice HTML di una pagina Web
	• Creare una semplice pagina HTML
	• Inserire titoli e sottotitoli
	• Inserire un paragrafo
	• Inserire una barra orizzontale
	• Creare elenchi puntati e numerati
	• Inserire una tabella
	• Creare un collegamento
	• Inserire un'immagine in una pagina
	• Inserire un video o un suono
	• Creare pagine con i form
	• Inserire fogli di stile in linea, incorporati, collegati
	• Creare una classe
	• Definire un identificatore
	• Validare l'accessibilità di un sito Web
	• Creare pagine HTML responsive e valutarne le caratteristiche di re-size automatico sui diversi dispositivi.

Periodo: aprile -giugno	

OBIETTIVI MINIMI

Conoscenze	Abilità
• 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	• 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

METODOLOGIE DIDATTICHE PREVISTE

- Lezione frontale e interattiva con l'uso della lim
- Apprendimento cooperativo attraverso la quale gli studenti apprendono in piccoli gruppi, aiutandosi reciprocamente e sentendosi corresponsabili del reciproco percorso
- Soluzione di problemi reali/ Problem solving che consente di analizzare, affrontare e cercare di risolvere positivamente situazioni problematiche. Obiettivi: trovare la soluzione e rendere disponibile una descrizione dettagliata del problema e del metodo per risolverlo.
- Brain-storming per consentire il confronto tra gli studenti e la valutazione di nuove idee.
- Gruppi di lavoro

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

STRUMENTI E ATTREZZATURE NECESSARIE ALLO SVOLGIMENTO DEL PERCORSO FORMATIVO

- libro di testo
  A. Lorenzi, V. Moriggia, Informatica per Istituti Tecnici Tecnologici, Volume A – Atlas
  ISBN 978-88-268-1838-2 pagine 480 € 19.60

- contenuti digitali integrativi del libro di testo
- materiali tratti da Internet
- laboratorio di Informatica con
  - *PC connessi ad Internet*
  - *Compilatore C++*
  - *Ambiente di sviluppo IDE per C++ (DEV C++)*
  - *Software Qt (qt-project.org)*
    "Software development made smarter
    Create fluid, high-performance and intuitive UIs, applications, and embedded devices – with the same code base for all platforms. "
    Sviluppo software multipiattaforma in C++ per dispositivi embedded & desktop.
  - *Start free Qt trial*
  - *Editor di pagine HTML, file CSS e script JavaScript*
- LIM

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 orali: interrogazioni, interventi significativi e partecipazione al dialogo educativo.

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

CRITERI DI VALUTAZIONE, INDICATORI E GRIGLIE

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