

**I.T.I. “E. MAJORANA” SOMMA VESUVIANA  
PIANO DI LAVORO ANNUALE DEL DOCENTE  
ANNO SCOLASTICO 2017/2018**

**Indirizzo:** Informatica e Telecomunicazioni  
**Articolazione:** Telecomunicazioni  
**Disciplina:** Informatica  
**Classe :** III sez. D  
**Docenti:** prof.ssa Buonaiuto Carmelina  
prof. Vicidomini Salvatore

### **Situazione di partenza**

La classe è composta da 24 allievi provenienti da sezioni differenti di cui 10 non ammessi alla classe quarta.

Prima di predisporre la programmazione didattica si è proceduto all'accertamento dei livelli di partenza dei singoli alunni con iniziali verifiche interattive in modo da ottenere elementi utili per decidere gli itinerari didattici. La situazione di partenza della classe è di una conoscenza sommaria dell'architettura e delle componenti di un computer ma di nessun fondamento di programmazione, né di alcun concetto di algoritmo (ad eccezione degli alunni ripetenti provenienti dalle classi di Informatica e Telecomunicazioni).

Il piano di lavoro dovrà pertanto tener conto di tali lacune che dovranno essere necessariamente sanate prima di affrontare gli argomenti previsti per la disciplina di Informatica del terzo anno (articolazione Telecomunicazioni). I tempi che si dedicheranno a ciascuna unità didattica saranno opportunamente tarati in itinere in base ai tempi di acquisizione degli allievi. Al fine di facilitare gli allievi nel raggiungimento degli obiettivi ci si fermerà spesso durante le lezioni frontali per fornire tutti i chiarimenti possibili ricorrendo all'uso di molti esempi, anche per illustrare le cose più semplici, ripetendo più volte durante l'esposizione i contenuti illustrati, facendo spesso lezioni di riepilogo interattive, effettuando diverse verifiche formative per rilevare con tempestività eventuali carenze nei processi di apprendimento al fine di attivare tempestivamente procedure compensative.

### **Obiettivi generali del corso**

Il corso si pone come obiettivi l'introduzione all'informatica di base portando pian piano gli allievi dapprima alla comprensione di problemi piuttosto semplici, rispetto ai quali imparano ad effettuare l'analisi e ad individuare una strategia risolutiva, con la relativa rappresentazione, fino all'utilizzo del codice come mezzo per comunicare con il computer e fare eseguire a questo le istruzioni che hanno individuato nella fase di analisi.

### **Competenze**

- › sapersi orientare nell'ambito della disciplina
- › affrontare e risolvere situazioni problematiche utilizzando le strategie del pensiero razionale negli aspetti dialettici e algoritmici
- › saper realizzare algoritmi e programmi in C++ per la soluzione di semplici problemi affrontando e documentando le diverse fasi di lavoro
- › saper usare un calcolatore per l'esecuzione dei propri algoritmi
- › saper costruire e rappresentare correttamente un algoritmo sia in pseudocodice che in un linguaggio di programmazione ricorrendo anche all'uso di sottoprogrammi
- › scegliere l'organizzazione dei dati più adatta a gestire le informazioni

## **Conoscenze**

- > Conoscere la fasi essenziali dello sviluppo del software
- > Il concetto di problema e le tecniche di risoluzione
- > Fasi di lavoro nella soluzione di un problema e loro documentazione
- > Significato di algoritmo
- > Tipi di dati
- > Comandi principali di Windows e dell'ambiente di sviluppo per il C++
- > Elementi di base della programmazione strutturata e loro rappresentazione in pseudocodice e nel linguaggio C++
- > La metodologia top-down
- > I dati strutturati

## **Abilità**

- > distinguere tra i concetti di problema e algoritmo, risolutore ed esecutore
- > pervenire alla soluzione di problemi utilizzando logiche elaborative
- > esprimere procedimenti risolutivi attraverso algoritmi
- > usare con proprietà un linguaggio di programmazione imperativo strutturato
- > documentare, a livello elementare, il lavoro svolto nella soluzione di un problema
- > scegliere l'organizzazione dei dati più adatta a gestire le informazioni

## **Metodi**

Lezione frontale per l'introduzione di argomenti teorici e della sintassi del linguaggio C++; lavoro di gruppo sia per la soluzione di problemi in classe che durante le esercitazioni pratiche. Verifiche individuali e di gruppo in classe e in laboratorio.

## **Strumenti**

Libro di testo, lavagna, dispense, appunti del corso.

## **Spazi**

Aula, laboratorio

## **Verifica**

Le verifiche formative saranno effettuate con continuità in un continuo dialogo con la classe mediante:

- Domande dal posto
- Correzione di esercizi assegnati

Per quanto riguarda le verifiche a carattere sommativo si utilizzeranno:

- Prove orali: interrogazioni, interventi significativi e partecipazione al dialogo educativo.
- Prove di laboratorio che prevederanno:
  1. esperienza e relazione scritta svolte individualmente
  2. discussione individuale di una relazione

Nelle verifiche di laboratorio si richiederà la soluzione di semplici problemi tramite l'implementazione di un programma in linguaggio C++ che dovrà essere documentato

opportunamente con analisi dei dati, analisi del problema, descrizione in metalinguaggio dell'algoritmo e testando il programma al computer.

Le verifiche di laboratorio saranno generalmente previste al termine di ciascuna unità didattica.

### **Valutazione**

La valutazione verrà fatta, in accordo con quanto stabilito dal consiglio di classe.

### **Contenuti e tempi didattici.**

In riferimento ai contenuti e all'indicazione dei tempi didattici le ore di lezione previste per l'anno scolastico 2017-2018 verranno così suddivise:

## **I QUADRIMESTRE**

**Settembre/Ottobre**

### **MODULO 1 : “INTRODUZIONE ALLO STUDIO DELL’INFORMATICA”**

#### **Unità didattica 1 : “ *Progettazione di algoritmi* ”**

- Concetto di algoritmo e sue caratteristiche
- Relazione tra algoritmo e programma
- Cos'è un linguaggio di programmazione
- Codice sorgente, assemblativo ed eseguibile
- I diversi linguaggi di programmazione
- Compilatori e interpreti
- Il ciclo di vita del software
- La programmazione strutturata

#### **Unità didattica 2: “*Programmare in C++*”**

- Il linguaggio C++
- Come si scrive un programma in C++
- Installazione e conoscenza dell'ambiente di sviluppo DEV-C++
- Alfabeto e regole lessicali in C++
- Input e output dei dati
- Il concetto di variabile e costante
- Istruzioni di comunicazione con l'utente
- Classificazione dei tipi di dati del C++ ( intero, reale, carattere e booleano)
- L'istruzione di assegnamento
- Gli operatori aritmetici
- Istruzioni in sequenza

#### **Abilità**

- Comprendere la relazione tra algoritmo e programma
- Acquisire il concetto di linguaggio di programmazione
- Conoscere gli strumenti a disposizione del programmatore per programmare

- Acquisire il concetto di programmazione strutturata
- Comprendere il concetto di variabile

### **Competenze**

- Saper utilizzare la terminologia informatica
- Saper individuare le diverse fasi della produzione del software
- Saper classificare i linguaggi nei diversi paradigmi di programmazione
- Saper editare e testare un programma
- Saper effettuare l'input e l'output dei dati
- Saper utilizzare le variabili e le costanti nei programmi

### **Novembre/Dicembre**

## **MODULO 2: “LA PROGRAMMAZIONE STRUTTURATA: LA SELEZIONE”**

### **Unità didattica 1: “La selezione semplice, doppia, nidificata”**

- Due tecniche di rappresentazione degli algoritmi: pseudocodifica e diagramma a blocchi
- Istruzioni in sequenza e commenti
- Istruzione di selezione semplice
- Istruzione di selezione doppia
- Operatori relazionali
- Operatori logici
- La selezione nidificata
- L'istruzione switch

### **Abilità**

- Codificare la selezione semplice e doppia
- Indentare il codice
- Utilizzare le variabili di tipo bool
- Utilizzare gli operatori logici e relazionali
- scrivere codice complesso con blocchi di istruzioni annidate

### **Competenze**

- Saper individuare le situazioni di selezione
- Saper concatenare le selezioni con le altre istruzioni
- Saper combinare più selezioni con condizioni logiche composte
- Saper riconoscere quando utilizzare selezioni annidate

### **Gennaio/Febrero**

## **MODULO 3: “LA PROGRAMMAZIONE STRUTTURATA: L'ITERAZIONE”**

### **Unità didattica 1: “L'iterazione precondizionata, postcondizionata e a conteggio”**

- Il ciclo a condizione iniziale : **WHILE**; utilizzo e rappresentazione in pseudolinguaggio e Diagramma a blocchi
- Utilizzo della trace table per verificare i programmi

- Il ciclo a condizione finale: **DO...WHILE**; utilizzo e rappresentazione in pseudolinguaggio e Diagramma a blocchi
- Il ciclo a conteggio **FOR** ; utilizzo e rappresentazione in pseudolinguaggio e Diagramma a blocchi

### **Abilità**

- Codificare programmi con il ciclo while
- Codificare programmi con il ciclo do ... while
- Codificare programmi con il ciclo for
- Utilizzare cicli annidati

### **Competenze**

- Saper trasformare le istruzioni di iterazione
- Saper scegliere il ciclo più adatto al problema da risolvere

## **IL QUADRIMESTRE**

### **Febbraio/Marzo**

#### **MODULO 3: “METODOLOGIA TOP – DOWN: LE FUNZIONI”**

##### **Unità didattica 1: “Funzioni e Procedure”**

- Metodologia di analisi: top-down e bottom-up
- Sottoprogrammi e sottoalgoritmi
- Visibilità ed ambienti di esecuzione: ambiente locale e globale
- Le procedure e le funzioni
- Passaggio di parametri per valore e per riferimento

### **Abilità**

- saper scrivere algoritmi utilizzando le funzioni
- saper utilizzare il meccanismo del passaggio dei parametri
- saper effettuare un passaggio dei parametri per valore e per riferimento
- saper distinguere l'ambiente locale da quello globale

### **Competenze**

- Saper decidere in quali casi è utile l'utilizzo di variabili globali
- Saper definire ed utilizzare funzioni personali

### **Aprile/Maggio**

#### **MODULO 4: “STRUTTURE DI DATI”**

##### **Unità didattica 1 : “I Vettori”**

- Array monodimensionale: i vettori
- Caricamento e scansione di un vettore

- Shift e rotazione di un vettore
- Ordinamento di un vettore
- Il problema della ricerca: ricerca sequenziale
- Passaggio a sottoprogrammi
- Le stringhe

### **Unità didattica 2 : “Le matrici”**

- Array a due dimensioni: le matrici
- Dichiarazioni di matrici
- Riempimento e stampa del contenuto di una matrice
- Passaggio a sottoprogrammi

#### **Abilità**

- Saper definire array monodimensionali
- Saper scrivere e leggere dati da un vettore
- Saper ordinare un vettore
- Saper ricercare un elemento in un vettore
- Saper operare con le stringhe
- Saper definire array bidimensionali
- Saper scrivere e leggere dati da una matrice

#### **Competenze**

- Saper manipolare i dati di un vettore
- Saper manipolare i dati di una matrice

**Somma Vesuviana, 13/11/2017**

#### **I Docenti**

Prof.ssa Buonaiuto Carmelina  
Prof. Vicidomini Salvatore