

## Istituto di Istruzione Superiore

“Di Vittorio - Lattanzio”

a.s. 2018 / 2019

### PROGETTAZIONE DIDATTICA

<b>MATERIA / DISCIPLINA</b>	<b>INFORMATICA</b>
<b>CLASSE</b>	<b>3 Bt LATTANZIO</b>
<b>DOCENTI</b>	Insegnante teorico: Prof. Tullio Testa  Insegnante tecnico pratico: Prof. Raffaele Grippa
<b>Libro di testo</b>	F. Formichi, G. Meini, <i>Informatica Seconda edizione per Telecomunicazioni - Algoritmi e linguaggio C/C++ - Pagine web con HTML e CSS</i> , (vol. 1), Ed. Zanichelli, ISBN 9788808626943

### PROFILO D'INGRESSO

La classe è formata da 14 alunni. La verifica iniziale ha mostrato varie incertezze e lacune nei concetti informatici posseduti e, da quanto si è osservato fino ad oggi, la classe prova solo un limitato interesse per la disciplina, che esprime anche con partecipazione discontinua alle singole attività didattiche.

### Unità di Apprendimento 1

**TITOLO:** Problemi, algoritmi, basi del linguaggio C++

#### Obiettivi Specifici di Apprendimento

Lo studente utilizza le strategie del pensiero razionale negli aspetti dialettici ed algoritmici per affrontare situazioni problematiche elaborando opportune soluzioni; è introdotto ad esprimere tali soluzioni utilizzando le caratteristiche fondamentali del linguaggio C++.

#### Contenuti

Dati e informazioni  
Problemi e algoritmi  
Fasi di risoluzione di un problema  
La rappresentazione degli algoritmi mediante diagrammi a blocchi  
Fasi di sviluppo di un programma  
Codice sorgente e codice eseguibile  
La struttura fondamentale di un programma in C++  
Tipi di dato  
Variabili e costanti  
Operatori  
Operazioni di input e output  
Ambiente di programmazione Dev-C++

<b>PERIODO</b>
Settembre/Ottobre/Novembre

COMPETENZE	ABILITÀ/CAPACITÀ	CONOSCENZE
Sviluppare la soluzione di un problema mediante algoritmi e codificarli in C++.	Analizzare correttamente problemi. Progettare algoritmi risolutivi. Implementare algoritmi scrivendo programmi sintatticamente corretti in linguaggio C++.	Relazioni fondamentali tra problemi, informazioni e linguaggi. Diagrammi a blocchi. Struttura di un programma in C++, caratteristiche principali di dati, istruzioni, operatori. Istruzioni di input/output.
Usare l'ambiente di progettazione algoritmica AlgoBuild. Usare l'ambiente di programmazione Dev-C++.	Attuare le fasi del lavoro di programmazione per codificare e verificare gli algoritmi.	Strumenti per lo sviluppo del software: interfaccia e funzionalità principali di AlgoBuild e Dev-C++.

<b>METODOLOGIA</b>
<p>Lezione dialogata e partecipata          Apprendimento cooperativo          Classe capovolta          Esercitazioni in classe, in laboratorio e a casa</p>

<b>STRUMENTI e MATERIALI</b>
<p>Computer, lavagna, libro di testo, materiali/strumenti digitali o cartacei</p>

<b>ATTIVITA' LABORATORIALI</b>
<p>Analisi dell'interfaccia grafica di un software per la progettazione di algoritmi e di un software per la programmazione in C++. Utilizzo degli stessi per la creazione di algoritmi e programmi facenti uso delle istruzioni fondamentali e delle caratteristiche basilari del linguaggio C++.</p>

<b>MONITORAGGIO, VERIFICA e VALUTAZIONE</b>		
TIPOLOGIA delle PROVE	PERIODO	CRITERI di VERIFICA
Verifiche formative orali e/o scritte e sommativa scritta semi-strutturata (domande chiuse a risposta multipla e domande aperte)	<i>Ottobre / Novembre</i>	Acquisizione del linguaggio specifico Acquisizione dei contenuti Capacità operative Metodo di lavoro Completezza dello svolgimento

## Unità di Apprendimento 2

**TITOLO:** Programmazione strutturata in C++

### Obiettivi Specifici di Apprendimento

Lo studente utilizza le strategie del pensiero razionale negli aspetti dialettici ed algoritmici per affrontare situazioni problematiche elaborando opportune soluzioni; è introdotto ad esprimere tali soluzioni utilizzando le strutture di controllo nel linguaggio C++.

### Contenuti

La programmazione strutturata  
Il costrutto sequenza  
Il costrutto selezione  
Il costrutto ripetizione  
Cicli infiniti e cicli interrotti  
Strutture di controllo annidate

### PERIODO

Novembre/Dicembre

COMPETENZE	ABILITÀ/CAPACITÀ	CONOSCENZE
Sviluppare la soluzione di un problema mediante algoritmi e codifica di essi in C++.	Analizzare correttamente problemi. Progettare algoritmi risolutivi. Confrontare algoritmi diversi per la soluzione dello stesso problema.	Strutture di controllo (anche con annidamento): sequenza, selezione, iterazione; codifica di esse.

### METODOLOGIA

Apprendimento cooperativo  
Lezione dialogata e partecipata  
Classe capovolta  
Esercitazioni in classe, in laboratorio e a casa

### STRUMENTI e MATERIALI

Computer, lavagna, libro di testo, materiali/strumenti digitali o cartacei

**ATTIVITA' LABORATORIALI**

Creazione di algoritmi e programmi facenti uso delle strutture di controllo fondamentali.

**MONITORAGGIO, VERIFICA e VALUTAZIONE**

<b>TIPOLOGIA delle PROVE</b>	<b>PERIODO</b>	<b>CRITERI di VERIFICA</b>
Verifiche formative e verifica sommativa scritta semi-strutturata (domande chiuse a risposta multipla e domande aperte)	<i>Novembre / Dicembre</i>	Acquisizione del linguaggio specifico Acquisizione dei contenuti Capacità operative Metodo di lavoro Completezza dello svolgimento

## Unità di Apprendimento 3

**TITOLO:** Funzioni in C++

### Obiettivi Specifici di Apprendimento

Lo studente utilizza le strategie del pensiero razionale negli aspetti dialettici ed algoritmici per affrontare situazioni problematiche elaborando opportune soluzioni; sviluppa applicazioni informatiche creando e gestendo opportune funzionalità.

### Contenuti

Funzioni della libreria matematica del C++  
Definizione di una funzione  
Invocazione di una funzione  
Passaggio dei parametri per valore e per riferimento

### PERIODO

Dicembre/Gennaio

COMPETENZE	ABILITÀ/CAPACITÀ	CONOSCENZE
Definire funzioni secondo particolari scopi nelle situazioni date. Saper scegliere opportunamente la modalità di passaggio dei parametri.	Progettare e implementare algoritmi utilizzando funzioni. Distinguere parametri attuali e formali. Usare funzioni predefinite.	Struttura di una funzione. Passaggio di parametri per valore e per riferimento. Regole di visibilità. Funzioni predefinite.

### METODOLOGIA

Lezione dialogata e partecipata  
Apprendimento cooperativo  
Classe capovolta  
Esercitazioni in laboratorio e a casa

### STRUMENTI e MATERIALI

Computer, lavagna, libro di testo, materiali/strumenti digitali o cartacei

**ATTIVITA' LABORATORIALI**

Creazione di algoritmi e programmi facenti uso di funzioni.

**MONITORAGGIO, VERIFICA e VALUTAZIONE**

<b>TIPOLOGIA delle PROVE</b>	<b>PERIODO</b>	<b>CRITERI di VERIFICA</b>
Verifiche formative e verifica sommativa scritta semi-strutturata (domande chiuse a risposta multipla e domande aperte) e/o verifica al computer	<i>Gennaio</i>	Acquisizione dei contenuti Capacità operative Completezza dello svolgimento Acquisizione del linguaggio specifico

## Unità di Apprendimento 4

**TITOLO:** Gli array

### Obiettivi Specifici di Apprendimento

Lo studente utilizza le strategie del pensiero razionale negli aspetti dialettici ed algoritmici per affrontare situazioni problematiche elaborando opportune soluzioni; sviluppa applicazioni informatiche creando e gestendo strutture dati opportunamente.

### Contenuti

Dichiarazione di un array  
Uso di array  
Array monodimensionali e bidimensionali  
Array come parametri di funzioni  
Stringhe di caratteri

### PERIODO

Febbraio/Marzo

COMPETENZE	ABILITÀ/CAPACITÀ	CONOSCENZE
Saper organizzare dati di tipi complessi. Realizzare programmi in linguaggio C++ facendo uso di strutture dati e relative funzioni scelte e/o progettate in base alla situazione data.	Definire array in C++ Scrivere / leggere dati su / da un array. Scegliere il tipo di organizzazione dei dati più adatto a gestire le informazioni in una situazione data. Progettare e implementare algoritmi utilizzando diverse strutture di dati e funzioni per gestirle.	Concetti di vettore di elementi e matrice di elementi. Struttura di un array. Tipologie di accesso ai dati. Passaggio di array come parametri.

### METODOLOGIA

Lezione dialogata e partecipata  
Apprendimento cooperativo  
Classe capovolta  
Esercitazioni in laboratorio e a casa

**STRUMENTI e MATERIALI**

Computer, lavagna, libro di testo, materiali/strumenti digitali o cartacei

**ATTIVITA' LABORATORIALI**

Creazione di algoritmi e programmi facenti uso di array e funzioni sugli array.

**MONITORAGGIO, VERIFICA e VALUTAZIONE**

<b>TIPOLOGIA delle PROVE</b>	<b>PERIODO</b>	<b>CRITERI di VERIFICA</b>
Verifiche formative e verifica sommativa scritta semi-strutturata (domande chiuse a risposta multipla e domande aperte) e/o verifica al computer	<i>Febbraio / Marzo</i>	Acquisizione dei contenuti Capacità operative Completezza dello svolgimento Acquisizione del linguaggio specifico



## Unità di Apprendimento 5

**TITOLO:** Algoritmi di ordinamento e ricerca

### Obiettivi Specifici di Apprendimento

Lo studente utilizza le strategie del pensiero razionale negli aspetti dialettici ed algoritmici per affrontare situazioni problematiche elaborando opportune soluzioni; sviluppa applicazioni informatiche creando e gestendo strutture dati opportunamente.

### Contenuti

Algoritmi per l'ordinamento di un array  
Algoritmi per la ricerca di un elemento in un array

### PERIODO

Marzo/Aprile

COMPETENZE	ABILITÀ/CAPACITÀ	CONOSCENZE
Classificare gli algoritmi di ordinamento e quelli di ricerca. Creare un'applicazione che ne faccia uso.	Codificare algoritmi di ordinamento e ricerca.	Algoritmi di ordinamento e loro caratteristiche. Algoritmi di ricerca e loro caratteristiche.

### METODOLOGIA

Apprendimento cooperativo  
Lezione dialogata e partecipata  
Classe capovolta  
Esercitazioni in classe, in laboratorio e a casa

### STRUMENTI e MATERIALI

Computer, lavagna, libro di testo, materiali/strumenti digitali o cartacei

**ATTIVITA' LABORATORIALI**

Creazione di algoritmi e programmi facenti uso di array e funzioni di ordinamento e ricerca sugli array.

**MONITORAGGIO, VERIFICA e VALUTAZIONE**

<b>TIPOLOGIA delle PROVE</b>	<b>PERIODO</b>	<b>CRITERI di VERIFICA</b>
Verifiche formative e verifica sommativa scritta semi-strutturata (domande chiuse a risposta multipla e domande aperte) e/o verifica al computer	<i>Aprile</i>	Acquisizione del linguaggio specifico Acquisizione dei contenuti Capacità operative Completezza dello svolgimento

## Unità di Apprendimento 6

**TITOLO:** Programmazione Web – Linguaggio HTML

### Obiettivi Specifici di Apprendimento

Lo studente sviluppa applicazioni informatiche per reti locali o servizi a distanza.

### Contenuti

Struttura di un documento HTML  
Elementi fondamentali del linguaggio HTML  
I collegamenti ipertestuali (link)  
Le immagini nelle pagine Web  
Suddivisione della pagina in funzione del contenuto

### PERIODO

Aprile/Maggio

COMPETENZE	ABILITÀ/CAPACITÀ	CONOSCENZE
Realizzare software per creare pagine Web statiche.	Progettare, realizzare e gestire pagine web statiche.	Linguaggi per la definizione delle pagine web: Elementi di linguaggio HTML. Programmazione lato client.

### METODOLOGIA

Apprendimento cooperativo  
Lezione dialogata e partecipata  
Classe capovolta  
Esercitazioni in laboratorio e a casa

### STRUMENTI e MATERIALI

Computer, lavagna, libro di testo, materiali/strumenti digitali o cartacei

**ATTIVITA' LABORATORIALI**

Esercitazioni con un software per la realizzazione di pagine Web statiche e con un browser per eseguirle e verificarne il funzionamento.

**MONITORAGGIO, VERIFICA e VALUTAZIONE**

<b>TIPOLOGIA delle PROVE</b>	<b>PERIODO</b>	<b>CRITERI di VERIFICA</b>
Verifiche formative e verifica sommativa scritta semi-strutturata (domande chiuse a risposta multipla e domande aperte) e/o verifica al computer	<i>Maggio</i>	Acquisizione del linguaggio specifico Acquisizione dei contenuti Capacità operative Completezza dello svolgimento

## Unità di Apprendimento 7

**TITOLO:** Aspetti di sicurezza informatica

### Obiettivi Specifici di Apprendimento

Lo studente viene condotto a gestire progetti secondo le procedure e gli standard previsti dai sistemi aziendali di gestione della qualità e della sicurezza; utilizza le reti e gli strumenti informatici nelle attività di studio, ricerca ed approfondimento disciplinare

### Contenuti

La sicurezza dei sistemi informatici  
La sicurezza nelle reti

### PERIODO

Febbraio/Maggio

COMPETENZE	ABILITÀ/CAPACITÀ	CONOSCENZE
Acquisire la consapevolezza dei vantaggi e dei limiti dell'uso degli strumenti e dei metodi informatici e delle conseguenze sociali e culturali di tale uso.	Applicare le normative di settore sulla sicurezza. Riconoscere i limiti e i rischi dell'uso della rete.	Problematiche relative alla sicurezza informatica. Tecniche di sicurezza in ambito aziendale. Normative di settore sulla sicurezza.

### METODOLOGIA

Apprendimento cooperativo  
Lezione dialogata e partecipata

### STRUMENTI e MATERIALI

Computer, lavagna, materiali digitali o cartacei

### ATTIVITA' LABORATORIALI

Ricerca di contenuti in rete.

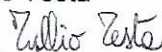
MONITORAGGIO, VERIFICA e VALUTAZIONE		
TIPOLOGIA delle PROVE	PERIODO	CRITERI di VERIFICA
Verifica sommativa scritta semi-strutturata (domande chiuse a risposta multipla e domande aperte)	Maggio	Acquisizione del linguaggio specifico Acquisizione dei contenuti Completezza dello svolgimento

Per gli obiettivi minimi si rimanda alla programmazione di dipartimento.

DATA 31/10/2018

FIRMA DOCENTI

Prof. Tullio Testa



Prof. Raffaele Grippa

