

# Programmazione didattica di Informatica

## Obiettivi e criteri di valutazione

### 1. **Obiettivi principali della disciplina**

L'insegnamento di informatica deve contemperare diversi obiettivi:

- comprendere i principali fondamenti teorici delle scienze dell'informazione,
- acquisire la padronanza di strumenti dell'informatica,
- utilizzare tali strumenti per la soluzione di problemi significativi in generale, ma in particolare connessi allo studio delle altre discipline,
- 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.

Questi obiettivi si riferiscono ad aspetti fortemente connessi fra di loro, che vanno quindi trattati in modo integrato. Il rapporto fra teoria e pratica va mantenuto su di un piano paritario e i due aspetti vanno strettamente integrati evitando sviluppi paralleli incompatibili con i limiti del tempo a disposizione.

Al termine del percorso liceale lo studente padroneggia i più comuni strumenti software per il calcolo, la ricerca e la comunicazione in rete, la comunicazione multimediale, l'acquisizione e l'organizzazione dei dati, applicandoli in una vasta gamma di situazioni, ma soprattutto nell'indagine scientifica, e scegliendo di volta in volta lo strumento più adatto. Ha una sufficiente padronanza di uno o più linguaggi per sviluppare applicazioni semplici, ma significative, di calcolo in ambito scientifico. Comprende la struttura logico-funzionale della struttura fisica e del software di un computer e di reti locali, tale da consentirgli la scelta dei componenti più adatti alle diverse situazioni e le loro configurazioni, la valutazione delle prestazioni, il mantenimento dell'efficienza.

L'uso di strumenti e la creazione di applicazioni deve essere accompagnata non solo da una conoscenza adeguata delle funzioni e della sintassi, ma da un sistematico collegamento con i concetti teorici ad essi sottostanti.

Il collegamento con le discipline scientifiche, ma anche con la filosofia e l'italiano, deve permettere di riflettere sui fondamenti teorici dell'informatica e delle sue connessioni con la logica, sul modo in cui l'informatica influisce sui metodi delle scienze e delle tecnologie, e su come permette la nascita di nuove scienze.

E' opportuno coinvolgere gli studenti degli ultimi due anni in percorsi di approfondimento anche mirati al proseguimento degli studi universitari e di formazione superiore. In questo contesto è auspicabile trovare un raccordo con altri insegnamenti, in particolare con matematica, fisica e scienze, e sinergie con il territorio, aprendo collaborazioni con università, enti di ricerca, musei della scienza e mondo del lavoro.

Dal punto di vista dei contenuti il percorso ruoterà intorno alle seguenti aree tematiche: architettura dei computer (AC), sistemi operativi (SO), algoritmi e linguaggi di programmazione (AL), elaborazione digitale dei documenti (DE), reti di computer (RC), struttura di Internet e servizi (IS), computazione, calcolo numerico e simulazione (CS), basi di dati (BD). “

### 2. Contenuti specifici e obiettivi per classe

#### 2.1. Classe 1°

Moduli	Contenuti
<b>MODULO 1:</b> Introduzione all'informatica	Informatica e computer. Le principali aree dell'informatica. Dato, informazione e conoscenza Le tecnologie dell'informazione e della comunicazione. Servizi e applicazione dell'ICT. Le professioni legate all'ICT. Le tappe della storia dell'informatica. Reale, virtuale e aumentato Prospettive di sviluppo tecnologico tra presente e futuro.
<b>MODULO 2:</b> Sistemi e modelli	Definizione generale di sistema. Classificazione dei sistemi Modello e prototipo. Classificazione dei modelli in fisici (analogici, iconici) e astratti (grafici, logici, matematici).
<b>MODULO 3:</b> Architettura del computer	Hardware, software e firmware. Caratteristiche e funzionalità dell'architettura hardware di un PC secondo uno schema a strati funzionali e secondo lo schema di von Neumann. I componenti di un computer: CPU e memoria centrale, clock, GPU, bus e porte di comunicazione, periferiche di I/O e memorie di massa. Schema gerarchico delle memorie. Tipi di computer e le loro proprietà.
<b>MODULO 4</b> Rappresentazione dei numeri	I sistemi di numerazione I sistemi di numerazione posizionale: decimale, binario, ottale, esadecimale. Conversioni di base Conversione in decimale. Conversione da decimale. La rappresentazione dei numeri interi Segno e modulo Complemento a due. I numeri floating point.
<b>MODULO 5:</b> Introduzione ai sistemi operativi e uso del computer	Tipi di software: di sistema e applicativo. Ruolo e funzionalità dei sistemi operativi. Installazione del sistema operativo e dei driver per i dispositivi. Configurazione iniziale e aggiornamento del sistema. Interfaccia a riga di comando e grafica basata sull'ambiente desktop. Il file system e le tipologie di file. Operare con icone e finestre e per gestire file e cartelle. Problematiche legali relative alle licenze, al diritto d'autore e alla privacy. Ergonomia e modalità per operare in sicurezza alla postazione di lavoro.

# Liceo Scientifico "A. Einstein" - Rimini

## Dipartimento di Informatica

<b>MODULO 6:</b> Elaborazione testi	Presentazione generale degli applicativi di office automation, i modelli di documento e le operazioni comuni a più software applicativi. Introduce l'elaborazione testi, gli elementi di un documento di testo e l'interfaccia di LibreOffice Writer o Google Documenti o Microsoft Word. La formattazione diretta tramite selezione e applicazione di una proprietà grafica e la modalità che prevede la creazione e l'applicazione a parti di un documento di stili di formattazione. Elementi che possono arricchire un documento quali oggetti grafici, formule matematiche e collegamenti per la creazione di ipertesti. Funzionalità avanzate quali la ricerca, il controllo ortografico e gli strumenti di collaborazione e per la revisione dei documenti.
<b>MODULO 7:</b> Foglio elettronico	Caratteristiche e funzionalità del foglio elettronico, soffermandosi sull'analisi What-if. Elementi e caratteristiche dell'interfaccia di LibreOffice Calc o Google Fogli o Microsoft Excel. Struttura e formato dei dati. I concetti di cella, intervallo, etichetta e valore. Le tecniche per selezionare celle, inserire dati, modificarli, copiarli e spostarli, per applicare formati ai numeri e per personalizzare tabelle con bordi e sfondi. Esempi ed esercizi per l'uso del foglio elettronico attraverso l'esecuzione di semplici calcoli con l'impiego di formule e guida nella conoscenza delle varie categorie di funzioni di tipo matematico, statistico, logico e di ricerca. Indicazioni per creare ed elaborare grafici. Percorsi per impostare i parametri di stampa e stampare il foglio di lavoro. Gestione di dati tramite ordinamento e filtro.
<b>MODULO 8:</b> Introduzione alla programmazione	Definizione generale di problema ed ruolo dei dati (variabili e costanti)- Concetto di algoritmo e le sue caratteristiche fondamentali. I linguaggi di programmazione ad alto e a basso livello. Le figure di risolutore, programmatore, esecutore e utente. La pseudocodifica e il flowchart per la descrizione dell'algoritmo. Caratteristiche del problem solving : scomposizione di un problema in sottoproblemi.
<b>MODULO 9:</b> Introduzione ai flowchart con Flowgorithm	Installazione di Flowgorithm e le tipologie di blocco disponibili. Il significato dell'equivalenza "Algoritmi + Strutture dati = Programmi". Le strutture di controllo sequenza e selezione.
<b>MODULO 10:</b> Cicli, matematica e statistica	Struttura di controllo di ripetizione, i cicli infiniti, i cicli precondizionali e postcondizionali. Classificazione degli errori di programmazione Debug Elementi di statistica e il calcolo del fattoriale.

### Educazione civica

Origini e evoluzione del computer e dell'informatica

Netiquette. Diritto d'autore e copyright

### Obiettivi essenziali

- Capire che cosa sono l'informatica e il computer
- Conoscere applicazioni e usi delle tecnologie dell'informazione e della comunicazione
- Conoscere le tappe e i principali protagonisti della storia dell'informatica
- Capire che cos'è un sistema e conoscerne la classificazione
- Capire l'utilità dei modelli per studiare e progettare i sistemi
- Conoscere i componenti di un sistema di elaborazione e il modello di Von Neumann
- Conoscere l'hardware e saper scegliere i dispositivi in base all'uso che se ne vuol fare
- Conoscere e saper utilizzare l'unità di misura della capacità di memoria
- Conoscere il sistema di numerazione binario e saper rappresentare con esso le informazioni.
- Saper convertire un numero da base 10 a base 2, 8, 16 e da base 2, 8, 16 a base 10.
- Saper rappresentare numeri naturali e numeri interi in segno e modulo e in complemento a due
- Conoscere la rappresentazione floating point
- Conoscere i tipi di software e le relative licenze
- Conoscere e saper eseguire le funzioni principali di un sistema operativo
- Conoscere le misure di prevenzione e di sicurezza da osservare quando si opera al PC
- Saper utilizzare le funzioni di base di un word processor
- Saper realizzare una presentazione multimediale
- Saper utilizzare le funzioni di base del foglio elettronico
- Conoscere i concetti di problema, algoritmo e variabile
- Conoscere le principali proprietà degli algoritmi
- Conoscere i concetti di linguaggio di programmazione e programma
- Descrivere graficamente semplici algoritmi con flowchart e pseudocodice
- Conoscere i principi base del problem solving
- Conoscere gli elementi di base di Flowgorithm
- obiettivo sostanzialmente raggiunto, con qualche imprecisione: sette
- obiettivo pienamente raggiunto: otto
- obiettivo raggiunto in maniera articolata, documentata, originale: nove o dieci

## 2.2. Classe 2°

Moduli	Contenuti
<b>MODULO 1:</b> Introduzione a Python. Uso della shell e dell'editor di Thonny	Ripasso degli argomenti dell'anno precedente, algoritmi e loro rappresentazione con Flowgorithm. Il linguaggio di programmazione Python. Perché studiare python, zen, storia ed ecosistema. Installazione dell'ambiente di sviluppo Thonny e uso dell'interfaccia. Il vocabolario di python, le 33 parole chiave(*)

# Liceo Scientifico "A. Einstein" - Rimini

## Dipartimento di Informatica

<b>MODULO 2:</b> Variabili e tipi di dato. Operatori matematici, espressioni, funzioni e moduli predefiniti (built-in)	Case sensitive. I commenti, le variabili ed i tipi di dato, il casting, nomi di variabili in notazione snake e Camel case. I contenitori liste[ ], range( ), tuple(), set{ }, dizionario{ }. Gli operatori matematici e le espressioni. L'assegnamento. Le funzioni predefinite. I moduli(math, random, turtle, tkinter) e package. Il modulo math. (*)
<b>MODULO 3:</b> La selezione, operatori logici e di confronto	La sequenza. La selezione semplice, doppia, in sequenza, annidata e multipla. Gli operatori logici nella selezione. Operatori logici e di comparazione(*)
<b>MODULO 4:</b> Iterazione	L'istruzione di iterazione. L'iterazione indefinita while( ) e definita for( ). Iterazioni annidate. (*). L'istruzione break e continue.
<b>MODULO 5:</b> Funzioni	La scomposizione in sottoprogrammi. Le funzioni in Python. La condivisione delle variabili nei sottoprogrammi. Le modalità di passaggio dei parametri alle funzioni, parametri attuali e formali. Le variabili locali e globali (*). Le funzioni anonime
<b>MODULO 6:</b> Tipi di dati composti: Le liste e le stringhe.	Operazioni con le liste/stringhe: concatenamento, parte di lista/stringa. Funzioni principali che operano sulle liste/stringhe: upper( ), find( ), replace(), ecc... (*)
<b>MODULO 7:</b> Tuple, set, dizionari	Dichiarazione di tipi di dati composti tuple, set, dizionari. Operazioni principali sui dati composti attraverso l'uso di funzioni e metodi a loro associati(*).
<b>MODULO 8:</b> Algoritmi notevoli	Algoritmi di ricerca , ordinamento e ricorsione

(\*) I contenuti essenziali sono quelli indicati con il simbolo asterisco di cui sopra.

### Educazione civica

**Ada Lovelace La prima programmatrice della storia**

**Video game e gioco d'azzardo**

### Obiettivi essenziali

- Saper fare l'analisi di un semplice problema
- Saper definire l'algoritmo risolutivo di un semplice problema e saperlo rappresentare
- Saper utilizzare l'ambiente Thonny per sviluppare semplici programmi in linguaggio Python
- Conoscere la struttura base di un programma nel linguaggio Python.
- Conoscere e saper rappresentare le variabili e funzioni
- Conoscere e saper utilizzare i tre schemi fondamentali della programmazione strutturata: sequenza, selezione e iterazione.
- Conoscere e saper utilizzare i tipi di dati composti

### 2.3. Classe 3°

Moduli	Contenuti
<b>MODULO 1:</b> Il web: Internet, documenti e linguaggi di markup	Storia e struttura di internet. Architettura client e server. Natura e rappresentazione di un documento, supporto e formato. Modalità di rappresentazione di un documento. Linguaggi markup ed i metadati. Il programma Visual Studio Code (*)
<b>MODULO 2:</b> Il web: Il linguaggio HTML	Il linguaggio HTML: tag ed attributi, strutturazione del testo, immagini, link, liste e tabelle. I tag contenitori : div e span. Creazione di un sito e pubblicazione sul web mediante servizio di hosting(*)
<b>MODULO 3:</b> Il web: Il linguaggio CSS	I fogli di stile a cascata CSS. Creazione, struttura ed uso delle regole di stile in una pagina web. L'attributo style, il tag style e il file style.css esterno. Gli attributi id e class. Tecniche per rendere un sito responsive(*)
<b>MODULO 4:</b> Il linguaggio Javascript	Pagine web dinamiche mediante il linguaggio javascript. Le variabili ed i tipi di dati semplici e composti. Le strutture di controllo e le funzioni. La gestione degli eventi (*) Le classi e gli oggetti. Debug tramite funzioni specifiche del browser
<b>MODULO 5:</b> Il linguaggio Python: gestione file ed elaborazione testi	Operazioni sui file: creazione, cancellazione, lettura e scrittura. Formato dei file: txt, csv, xml e json. Analisi delle frequenze e grafici con libreria Python
<b>MODULO 6:</b> Creazione di un sito web	Siti web statici e dinamici. Layout e griglie. Contenuti, pagine e menu di navigazione. Creazione di un server web tramite la libreria python flask ed installazione su server Linux Debian. I CMS e creazione di un sito web utilizzando il CMS WordPress

(\*) I contenuti essenziali sono quelli indicati con il simbolo asterisco di cui sopra.

#### Obiettivi essenziali

- Saper fare l'analisi di un problema web proposto
- Saper realizzare un layout per una pagina web
- Saper utilizzare l'ambiente Visual Studio Code per sviluppare siti web statici e dinamici
- Conoscere la struttura base di una pagina web
- Conoscere e saper rappresentare le variabili e funzioni in Javascript e Pyscript

# Liceo Scientifico "A. Einstein" - Rimini

## Dipartimento di Informatica

---

- Conoscere e saper utilizzare i tre linguaggi principali del web HTML, CSS e Javascript.
- Conoscere e saper utilizzare i CMS per creare siti e saperli pubblicare su internet
- Conoscere e saper utilizzare il sistema operativo Linux versione Debian

### 2.4. Classe 4°

Moduli	Contenuti
<b>MODULO 1:</b> Introduzione alla programmazione ad oggetti	Panoramica sui principali paradigmi di programmazione e introduce la programmazione a oggetti in Python, partendo dai concetti di oggetto e messaggio. Illustra i principi fondamentali della programmazione a oggetti quali: l'incapsulamento, l'astrazione, l'ereditarietà e il polimorfismo. Inoltre, mostra come rappresentare le classi e gli oggetti utilizzando i rispettivi diagrammi in Unified Modelling Language (UML).
<b>Modulo 2:</b> Database e modello concettuale entità associazione	Le caratteristiche generali degli archivi di dati e dei sistemi informativi, il formato CSV e i limiti dei flat file legati alle anomalie nelle operazioni di inserimento, modifica e cancellazione; partendo da questi problemi, vengono introdotti i database, i concetti base e le tipologie principali. Il modello concettuale entità-associazione (ER) per progettare sistemi informativi con i suoi costrutti base: associazioni, attributi e identificatori. Le generalizzazioni e le gerarchie di entità.
<b>Modulo 3:</b> Progettazione logica e modello relazionale	La progettazione logica e il modello relazionale con i suoi costrutti fondamentali quali le tabelle, gli attributi, i vincoli intrarelazionali e il vincolo inter-relazionale di integrità referenziale. Le varie tipologie di chiave e l'uso del valore nullo. Tradurre uno schema concettuale ER nel corrispondente schema logico relazionale e la normalizzazione di uno schema relazionale.
<b>Modulo 4:</b> SQL e algebra relazionale	Le tipologie dei linguaggi per database. I principi base del linguaggio SQL, il DBMS Schemi DDL (Data Definition Language) I comandi DML (Data Manipulation Language). Interrogazioni sia in algebra relazionale sia in SQL. Funzioni di aggregazione

#### Educazione civica

Big data e open data  
Curriculum vitae on line

#### Obiettivi in termini di competenze, conoscenze e abilità

(gli obiettivi minimi sono preceduti dal simbolo [x]).

COMPETENZE

- [x] Applicare i principi della programmazione a oggetti nella programmazione.
- [x] Descrivere classi e oggetti tramite i relativi diagrammi UML.
- Scrivere programmi che prevedono associazioni tra classi.
- Comprendere il limite dei flat file come i file CSV.
- Distinguere le tipologie principali di database.
- [x] Tradurre uno schema concettuale entità-associazione nel corrispondente schema logico relazionale.
- Normalizzare in terza forma normale un database relazionale.
- [x] Creare lo schema di un database.
- Manipolare i dati di un database.
- [x] Interrogare una base di dati relazionale tramite il linguaggio SQL.
- Confrontare interrogazioni di un database con l'algebra relazionale e l'SQL.
- Recuperare dati contenuti in un database relazionale, utilizzando funzioni di aggregazione.

### CONOSCENZE DISCIPLINARI / OBIETTIVI COGNITIVI

- [x] I principali paradigmi di programmazione.
- [x] Principi della programmazione a oggetti (incapsulamento, astrazione, ereditarietà e polimorfismo).
- [x] Diagrammi UML delle classi.
- Diagrammi UML degli oggetti.
- [x] Associazioni UML tra classi.
- Archivi di dati e sistemi informativi.
- Flat file e formato CSV e anomalie nei dati.
- [x] Tipologie principali di database.
- Progettazione e tipi di database.
- [x] Modello entità-associazione (ER): entità, associazioni, attributi e identificatori.
- Generalizzazioni e gerarchie di entità.
- Modello logico relazionale.
- Vincoli di integrità intrarelazionali: di dominio, di chiave primaria, di tupla, di univocità e di opzionalità (valore nullo).
- Vincolo inter relazionale di integrità referenziale.
- Normalizzazione.
- [x] Linguaggio SQL.
- Comandi DDL e DML.
- [x] Algebra relazionale.
- Comando DQL per interrogazioni.
- Funzioni di aggregazione.

### ABILITÀ / OBIETTIVI OPERATIVI

- [x] Scrivere programmi Python definendo classi e istanziando oggetti.
- Scrivere programmi Python applicando l'incapsulamento.
- Scrivere programmi Python applicando ereditarietà e polimorfismo.
- [x] Disegnare il diagramma UML di una classe.
- Disegnare il diagramma UML di un oggetto.
- [x] Disegnare associazioni UML tra classi e implementare le associazioni in Python.
- Utilizzare l'OOP per disegnare figure geometriche.
- Individuare le anomalie presenti in un flat file.
- [x] Disegnare diagrammi ER con entità e associazioni per modellare un sistema informativo.
- Disegnare gerarchie di entità legate da generalizzazioni.

# Liceo Scientifico "A. Einstein" - Rimini

## Dipartimento di Informatica

---

Scrivere il modello logico relazionale di un database.

Applicare le regole di traduzione di uno schema concettuale ER nel corrispondente schema logico relazionale.

Normalizzare uno schema relazionale in prima, seconda e terza forma normale.

[x]Scrivere semplici interrogazioni SQL con il comando DQL SELECT.

[x]Scrivere interrogazioni con l'algebra relazionale.

Scrivere interrogazioni SQL con funzioni di aggregazione.

### 2.5. Classe 5°

Moduli	Contenuti
<b>MODULO 1:</b> Il linguaggio Python	Struttura di un programma in Python. Tipi di dato semplici. Le funzioni predefinite. La funzione random(). Sequenza, selezione e iterazione in Python. Funzioni. Tipi di dato composti. OOP in Python. Classi e oggetti. Ereditarietà. Il modulo py5
<b>MODULO 2:</b> Informatica teorica e modelli computazionali	I modelli computazionali. Gli automi a stati finiti Simulazione in Python di un automa a stati finiti. Gli automi cellulari. La macchina di Turing I frattali
<b>MODULO 3:</b> Complessità computazionale	Introduzione alla complessità computazionale. Analisi del costo computazionale. Algoritmi di ordinamento. I problemi intrattabili.
<b>MODULO 4:</b> Strategie di problem solving	Strategie e tattiche. Il problema delle otto regine. Il problema dello zaino
<b>MODULO 5:</b> Calcolo numerico e algebra lineare	Interfaccia JupyterLab La libreria NumPy Introduzione al calcolo numerico. Calcolo del valore di pi greco. Calcolo degli zeri di una funzione. Metodi di integrazione numerica per il calcolo delle aree.
<b>MODULO 6:</b> Intelligenza artificiale e retineurali	Introduzione all'intelligenza artificiale. Le tappe fondamentali dell'intelligenza artificiale. Il modello simbolico e sub-simbolico. Le reti neurali artificiali
<b>Modulo 7:</b> Data science	Introduzione ai big data e alla data science. L'analisi esplorativa dei dati e il data cleaning. La visualizzazione dei dati. Modelli di machine learning. Simulazione dei sistemi.
<b>Modulo 8:</b> Teoria dell'informazione e reti di calcolatori	Fondamenti di teoria dell'informazione I grafi. L'algoritmo di Dijkstra per trovare il cammino minimo in un grafo. Introduzione alle reti di calcolatori

<b>Modulo 9:</b> Internet e servizi di rete	La rete internet. Il modello ISO/OSI. La suite TCP/IP. I servizi di internet. Le applicazioni web con PyScript
<b>Modulo 10:</b> Sicurezza informatica	Introduzione alla sicurezza informatica I malware e gli attacchi informatici. Crittografia simmetrica e asimmetrica

### Educazione civica

*Firma elettronica, digitale, certificati, PEC e SPID*

### Obiettivi in termini di competenze, conoscenze e abilità

(gli obiettivi minimi sono preceduti dal simbolo [x]).

[x]Conoscere il ruolo dei modelli computazionali e dei linguaggi formali nell'informatica teorica

[x]Modellare un semplice sistema con un automa a stati finiti e simulare il suo comportamento

Comprendere i principi degli automi cellulari e simulare il gioco della vita di Conway

[x]Conoscere e implementare il modello computazionale della macchina di Turing

Simulare il sistema di riscrittura degli L-system per generare forme frattali

[x]Conoscere i fondamenti della teoria della complessità computazionale

[x]Determinare l'efficienza degli algoritmi in termini di tempo e di spazio

[x]Descrivere gli aspetti principali della teoria della complessità computazionale fornendo opportuni casi di studio

Riconoscere e spiegare le principali classi di complessità computazionale

Definire il concetto di calcolabilità e i limiti degli algoritmi

[x]Distinguere le principali strategie di problem solving in ambito informatico

[x]Esemplificare le strategie di problem solving classificando algoritmi e metodi risolutivi

Realizzare programmi per calcolare combinazioni, permutazioni e insieme delle parti

Risolvere problemi combinatori scegliendo la strategie più opportune

[x]Comprendere i principi fondamentali del calcolo numerico e dell'algebra lineare

[x]Adoperare i notebook Jupyter per il calcolo interattivo e la visualizzazione dei risultati

Utilizzare la libreria NumPy per manipolare array e matrici

[x]Applicare metodi numerici per il calcolo degli zeri di una funzione e per l'integrazione numerica

Implementare concetti di algebra lineare per effettuare operazioni con le matrici e risolvere sistemi di equazioni lineari

[x]Conoscere le tappe fondamentali dell'intelligenza artificiale (IA)

[x]Distinguere l'intelligenza artificiale simbolica dall'intelligenza artificiale sub-simbolica

[x]Conoscere e descrivere obiettivi e risultati dei principali modelli di intelligenza artificiale

Realizzare semplici reti neurali che computano porte logiche e una rete neurale per il riconoscimento delle cifre

[x]Conoscere i concetti base sui big data e la data science  
Analizzare dataset ed effettuare semplici operazioni di data cleaning  
Visualizzare dataset con vari tipi di diagramma  
Applicare modelli di machine learning sia supervisionati sia non supervisionati  
Scrivere programmi per la simulazione di sistemi biologici

[x]Spiegare i principi fondamentali della teoria dell'informazione  
Implementare metodi di compressione dei dati  
Descrivere la teoria dei grafi e il suo ruolo nella gestione delle reti  
[x]Identificare e descrivere le tipologie principali di reti di calcolatori  
[x]Descrivere i dispositivi di rete e il loro impiego nelle reti di calcolatori

[x]Confrontare il modello ISO/OSI e la suite TCP/IP per la progettazione e la realizzazione delle reti informatiche e di Internet  
Scrivere semplici programmi client-server e di web scraping  
Scrivere web app con PyScript

[x]Conoscere i concetti base della sicurezza informatica e le principali tipologie di malware e di attacco informatico  
Realizzare semplici programmi di attacco di password  
[x]Conoscere i principi della crittografia simmetrica e asimmetrica

### 3. Metodologia e strumenti utilizzati

Ogni modulo è articolato nel seguente modo:

- descrizione degli obiettivi;
- proposta di un problema attinente alla vita reale relativo all'argomento
- analisi del problema e individuazione di una soluzione che richieda l'uso degli
- strumenti oggetto di studio
- eventuale ripasso dei prerequisiti;
- spiegazioni teoriche e guida delle attività pratiche;
- prima verifica e analisi dei risultati;
- recupero e/o approfondimenti;
- verifica

Le spiegazioni teoriche sono ampiamente supportate da una didattica mirata al problem solving e alla ricerca guidata della soluzione dei problemi proposti.

L'attività di laboratorio, essenziale al completamento dell'apprendimento delle nozioni teoriche, viene considerata fondamentale ed integrata nell'ambito della stessa teoria con cui procede in parallelo.

**Libro di testo:**

**Hashtag 1(primo biennio), Boscaini-Montresor-Masetti, HOEPLI**

**Hashtag 2(secondo biennio), Boscaini-Montresor-Masetti, HOEPLI**

**Hashtag 3(quinta), Boscaini-Montresor-Masetti, HOEPLI**

Il libro di testo, il computer e la rete Internet saranno il punto di partenza per attività concretamente realizzate dagli studenti con eventuali altre risorse.

### 4. Prove di verifica

Per ogni periodo didattico si effettuano almeno due verifiche: accertamento dei prerequisiti, verifica formativa in itinere, verifica sommativa con modalità diverse a seconda dell'argomento trattato. Eventuali recuperi saranno gestiti preferibilmente in itinere, anche assegnando dei lavori da svolgere a casa controllati successivamente dall'insegnante.

In modo costante avviene anche l'osservazione degli alunni in relazione alle modalità di interazione nel gruppo, alla capacità di proporre soluzioni in modo chiaro e di ascoltare il punto di vista dei compagni, alla puntualità nelle consegne e al rispetto delle regole scolastiche.

Le tecniche utilizzate sono:

- Brainstorming
- Test scritti, preferibilmente a risposta aperta/ a risposta multipla/ vero-falso/ a completamento
- Interrogazioni orali
- Svolgimento di esercizi scritti
- Elaborazione di documenti in base a specifiche assegnate
- Attività pratiche di laboratorio

Esposizione di lavori svolti in laboratorio o ricerche individuali

### 5. Criteri di valutazione e scala valutativa

Tutte le discipline hanno un voto unico finale sia per il primo che il secondo periodo, con scala valutativa da 2 a 10.

La valutazione finale dei singoli periodi seguirà le indicazioni fornite dal Collegio dei Docenti.

Il docente individua di volta in volta l'opportunità di somministrare prove scritte, orali, pratiche o strutturate, secondo l'esigenza.

Nella valutazione finale si terrà conto dei seguenti aspetti:

- impegno e partecipazione;
- conoscenze (sapere);
- competenze (saper fare);
- capacità di rielaborazione logiche e critiche;
- abilità espressive;
- originalità.

### 6. Interventi individualizzati per il recupero e l'approfondimento

Si conviene che il criterio per individuare la necessità o meno di interventi di recupero sia la presenza di un numero significativo di insufficienze per quanto riguarda il singolo alunno, o di un numero elevato di esse in una verifica collettiva per quanto riguarda l'intera classe.

L'attività di recupero si effettuerà principalmente in itinere, cioè con ripresa di argomenti e proposta di esercizi riepilogativi nelle ore curricolari. Dopo il primo periodo si valuterà se effettueranno una/due settimane di rallentamento con verifica finale.

### 7. Criteri e metodi per la valutazione degli alunni con DSA

Per gli alunni con certificazione DSA/H le prove orali/scritte/pratiche e la loro valutazione avverranno in accordo a quanto stabilito nel PDP/PEI.

# Liceo Scientifico "A. Einstein" - Rimini

## Dipartimento di Informatica

### 8. Griglia di valutazione

GRIGLIA DI VALUTAZIONE PER PROVA SCRITTA, PRATICA ED ORALE					
COMPETENZE		CONOSCENZE	ABILITÀ	GIUDIZIO	VOTO
NON ACQUISITE	Nessuna trattazione o rifiuto di sottoporsi a verifiche scritte, pratiche ed orali	Conoscenze nulle o molto scarse dei contenuti di base	Incapacità di applicare qualsiasi procedimento risolutivo. Incapacità nell'esposizione orale.	Gravemente insufficiente	$2 \leq V \leq 3$
	Trattazione lacunosa e confusa. Errori logici gravi nell'applicazione delle conoscenze	Conoscenze lacunose sui contenuti di base	Incapacità di cogliere relazioni tra concetti di base. Gravi errori di procedimento su parti essenziali	Scarso	$3 < V < 5$
LIVELLO BASE	Trattazione incompleta che evidenzia una parziale comprensione dei concetti essenziali	Conoscenza incompleta e superficiale, difficoltà di collegamento tra i concetti.	Svolgimento incompleto e/o con errori non gravi. Esposizione incompleta e/o mnemonica con una parziale conoscenza del linguaggio specifico	Mediocre	$5 \leq V < 6$
	Possesso di conoscenze ed abilità essenziali. Risoluzione di problemi in situazioni note.	Conoscenza e comprensione dei nuclei concettuali della disciplina, formulati in modo corretto anche se poco approfondito.	Applicazione corretta di regole e procedure. Uso di un linguaggio semplificato ma corretto.	Sufficiente	$6 \leq V < 7$
LIVELLO INTERMEDIO	Risoluzione corretta di problemi complessi in situazioni note utilizzando consapevolmente le conoscenze ed abilità acquisite e valutando l'attendibilità dei risultati	Conoscenza ampia ed approfondita. Comprensione completa dei contenuti. Sufficienti capacità di rielaborazione personale.	Applicazione corretta e consapevole di regole e procedure anche in problemi più complessi. Esposizione con uso di un linguaggio appropriato	Discreto / Buono	$7 \leq V \leq 8$
LIVELLO AVANZATO	Svolgimento autonomo di problemi complessi in situazioni anche non note. Padronanza nell'uso delle conoscenze ed abilità	Conoscenza completa dei contenuti, rielaborata ed approfondita in modo personale	Capacità di applicare correttamente le conoscenze anche in situazioni nuove. Buone capacità espositive, utilizzo di un linguaggio specifico chiaro e rigoroso	Ottimo / Eccellente	$8 < V \leq 10$