

PROGRAMMAZIONE DI DIPARTIMENTO

**INFORMATICA**

prof. *Alessandro Zeroni*

**PROFILO GENERALE E COMPETENZE**

L'insegnamento dell'Informatica deve ottemperare diversi obiettivi:

- comprendere i fondamenti teorici delle scienze dell'informazione;
- acquisire la padronanza degli strumenti hardware/software per la risoluzione di problemi in chiave multidisciplinare;
- acquisire la consapevolezza delle potenzialità e dei limiti degli strumenti informatici anche in chiave sociale.

La libertà, la competenza e la sensibilità dell'insegnante svolgeranno un ruolo fondamentale nel proporre problemi significativi, strumenti e soluzioni aggiornate, tali da permettere il collegamento con le altre discipline. E' un dovere coinvolgere gli studenti del quarto e del quinto anno in percorsi di approfondimento mirati sia al proseguimento degli studi universitari sia all'accesso al mondo del lavoro.

Si decide così di raccordare l'informatica con la matematica, la fisica, le scienze e rafforzare le sinergie con il territorio, le collaborazioni con università, enti di ricerca e musei.

Dal punto di vista dei contenuti il percorso ruoterà attorno 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).

**OBIETTIVI SPECIFICI DI APPRENDIMENTO**

Primo e secondo anno: gli studenti apprenderanno i concetti fondamentali relativi alla codifica dell'informazione digitale, utilizzeranno gli strumenti hardware/software più comuni verificando settimanalmente in attività di laboratorio le proprie conoscenze.

- AC: architettura dell'elaboratore, concetti di hardware e software, codifica binaria, codici ASCII/UNICODE, modello di Von Neumann;
- SO: funzionalità di base dei moderni sistemi operativi, concetto di programma, processo, gestione della memoria, file system;
- DE: strumenti di produzione di fogli elettronici con particolare attenzione ai fogli di calcolo (Excel, Calc);
- AL: concetto e codifica di algoritmi, linguaggi di programmazione.

Terzo e quarto anno: verranno introdotti nuovi linguaggi di programmazione ed affrontati problemi di complessità maggiore in chiave multidisciplinare; nell'attività di laboratorio gli studenti utilizzeranno strumenti informatici per la costruzione di database e pagine web.

- RC: struttura di un rete di computer;
- IS: servizi rischi, legislazione della Rete Globale, ricerca delle informazioni;
- DB: introduzione al modello relazionale dei dati, linguaggi di interrogazione e manipolazione dei dati;
- DE: strumenti avanzati di produzione dei documenti elettronici, linguaggi di markup (XML), formati non testuali (bitmap, vettoriale, formati di compressione), font tipografici, progettazione web;

Quinto anno: l'insegnante valuterà di volta in volta il percorso didattico più adeguato alla singola classe, realizzando approfondimenti anche in raccordo con le altre discipline.

- CS: teoria dell'informazione (Shannon), algoritmi di compressione, complessità computazionale;
- IS: protocolli di rete;
- AL: programmazione orientata agli oggetti (JAVA);
- Elementi di crittografia.

**OBIETTIVI DEL PERCORSO FORMATIVO & OBIETTIVI MINIMI (sottolineati)**

<b>CLASSI</b>	<b>Conoscenze</b>	<b>Competenze</b>
<b>PRIME</b>		
Concetti Informatici di Base	<u>Concetti generali</u> <u>Sistemi di numerazione decimale e binario</u> <u>I connettivi logici</u> <u>Struttura generale del sistema di elaborazione</u> <u>Unità centrale di elaborazione</u> <u>Le memorie</u> <u>La codifica delle informazioni nella memoria</u> <u>La rappresentazione delle informazioni alfanumeriche, sonore, video</u> <u>Le unità di input e di output</u> <u>Le immagini digitali</u> Le memorie di massa Il collegamento delle periferiche Software e Licenze software	Saper convertire da base decimale a binaria e viceversa Saper riconoscere e classificare i componenti di un sistema di elaborazione Saper classificare le licenze software e comportarsi in accordo con esse
Sistema Operativo (Windows)	Caratteristiche generali <u>Avvio e arresto del sistema</u> <u>L'interfaccia standard delle applicazioni</u> <u>Il pulsante Start</u> <u>Il programma per l'editing dei testi</u> <u>Cartelle e file</u> <u>Le operazioni sui file</u> <u>La Guida in linea</u>	Saper utilizzare i componenti di un sistema di elaborazione Saper gestire file e cartelle Saper riconoscere i principali tipi di files Saper utilizzare stampante, hard disk e memorie esterne Saper utilizzare gli strumenti di supporto
Reti, Web e Comunicazione	<u>World Wide Web</u> <u>Browser</u> <u>Motore di ricerca</u> <u>Le reti nella vita di tutti i giorni</u> <u>Lavorare in rete</u>	Saper utilizzare correttamente il browser Saper effettuare ricerche articolare usando i motori di ricerca Saper utilizzare in maniera sicura le risorse di rete
Il foglio elettronico (MS Excel)	<u>Utilizzo del foglio di calcolo</u> <u>Principali funzioni</u> <u>Formattazione, Stampa</u>	Saper realizzare fogli di calcolo scrivendo formule anche complesse Utilizzare e combinare funzioni predefinite
Algoritmi e linguaggi	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 La programmazione strutturata Le strutture di controllo Teorema di Böhm-Jacopini	Tradurre semplici problemi in sequenze ordinate Saper tradurre sequenze ordinate in pseudocodici Saper utilizzare le principali strutture di controllo
<b>SECONDE</b>		
Algoritmi e linguaggi	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 La programmazione strutturata Le strutture di controllo Teorema di Böhm-Jacopini	Tradurre semplici problemi in sequenze ordinate Saper tradurre sequenze ordinate in pseudocodici Saper utilizzare le principali strutture di controllo
Linguaggio di programmazione Turbo Pascal	<u>Struttura generale di un programma Pascal</u> <u>Tipi di dati numerici e non numerici</u> <u>Dichiarazione delle costanti e delle variabili</u> <u>Istruzione di assegnazione e operatori</u> <u>Istruzioni di I/O</u> Fasi del lavoro di programmazione Errori sintattici, lessicali e logici Codifiche delle tre strutture fondamentali: sequenza, alternativa, ripetizione	<u>Scrivere un programma Pascal sintatticamente corretto</u> <u>Scegliere il tipo di dato adatto a rappresentare le variabili</u> <u>Validare un programma</u> <u>Produrre programmi documentati</u> Individuare le strutture di controllo più idonee per la soluzione di un problema Rappresentare la selezione Annidare strutture di controllo
Linguaggio di programmazione C++	<u>Struttura generale di un programma C++</u> <u>Tipi di dati numerici e non numerici</u> <u>Dichiarazione delle costanti e delle variabili</u> <u>Istruzione di assegnazione e operatori</u> <u>Istruzioni di I/O</u> Codifiche delle tre strutture fondamentali:	<u>Scrivere un programma C++ sintatticamente corretto</u> <u>Scegliere il tipo di dato adatto a rappresentare le variabili</u> <u>Validare un programma</u> <u>Produrre programmi documentati</u>

	<u>sequenza, alternativa, ripetizione</u> Strutture annidate di alternativa <u>Ripetizione precondizionale e ripetizione con contatore</u> Struttura di scelta multipla	<u>Individuare le strutture di controllo più idonee per la soluzione di un problema</u> <u>Rappresentare la selezione</u> <u>Annidare strutture di controllo</u> <u>Esaminare un elenco di dati</u> <u>Rappresentare le strutture derivate di ripetizione</u> <u>Calcolare il valore massimo tra un insieme di valori</u> <u>Utilizzare la struttura di scelta multipla</u>
<b>TERZE</b>		
Linguaggio C++	<u>Funzioni</u> <u>Funzioni con parametri</u> <u>Passaggio di parametri per referenza e per valore</u> Dichiarazione dei prototipi di funzione <u>Definizione di risorse locali e globali</u> <u>Regole di visibilità</u> Funzioni predefinite del linguaggio Namespace e librerie di inclusione Function overloading Funzioni ricorsive <u>Enumerazioni</u> <u>Array</u> Array a due dimensioni Strutture Puntatori	<u>Sviluppare un programma introducendo le funzioni</u> <u>Utilizzare il passaggio di parametri per referenza e per valore</u> Dichiarare le funzioni con i prototipi <u>Individuare le applicazioni pratiche delle regole di visibilità</u> Utilizzare le funzioni predefinite Ridefinire una funzione con overloading Individuare alcuni casi semplici di utilizzo di funzioni ricorsive <u>Creare un'enumerazione</u> <u>Organizzare i dati in array a una o due dimensioni</u> Organizzare i dati in strutture Utilizzare i puntatori
Programmazione ad oggetti	<u>Astrazione</u> <u>Programmazione procedurale e strutturata</u> <u>Classi, attributi e metodi</u> <u>Incapsulamento</u> <u>Costruttore e distruttore</u> <u>Il costruttore di copia</u> <u>Accesso di tipo public e private alle member function</u> Messaggi e interfaccia Funzioni di tipo friend <u>Ereditarietà</u> <u>Gerarchia delle classi</u> Polimorfismo Membri protetti Overloading dei metodi Overriding dei metodi Overloading degli operatori delle classi Funzioni virtuali Vantaggi della programmazione ad oggetti	<u>Applicare i principi della programmazione ad oggetti utilizzando il C++</u> <u>Creare una classe con attributi e metodi</u> <u>Dichiarare le istanze di una classe</u> <u>Definire un costruttore ed un distruttore di una classe</u> <u>Definire i costruttori di copia</u> Utilizzare le funzioni friend <u>Creare una classe derivata</u> <u>Saper utilizzare le clausole public e private</u> <u>Proteggere i membri di una classe</u> Applicare l'overloading ai metodi delle classi derivate Saper riscrivere i metodi delle classi derivate Applicare l'overloading agli operatori nelle classi Individuare le funzioni da rendere virtuali
<b>QUARTE</b>		
Programmazione ad oggetti	<u>Astrazione</u> <u>Programmazione procedurale e strutturata</u> <u>Classi, attributi e metodi</u> <u>Incapsulamento</u> <u>Costruttore e distruttore</u> <u>Il costruttore di copia</u> <u>Accesso di tipo public e private alle member function</u> Messaggi e interfaccia Funzioni di tipo friend <u>Ereditarietà</u> <u>Gerarchia delle classi</u> Polimorfismo Membri protetti Overloading dei metodi Overriding dei metodi Overloading degli operatori delle classi Funzioni virtuali Vantaggi della programmazione ad oggetti	<u>Applicare i principi della programmazione ad oggetti utilizzando il C++</u> <u>Creare una classe con attributi e metodi</u> <u>Dichiarare le istanze di una classe</u> <u>Definire un costruttore ed un distruttore di una classe</u> <u>Definire i costruttori di copia</u> Utilizzare le funzioni friend <u>Creare una classe derivata</u> <u>Saper utilizzare le clausole public e private</u> <u>Proteggere i membri di una classe</u> Applicare l'overloading ai metodi delle classi derivate Saper riscrivere i metodi delle classi derivate Applicare l'overloading agli operatori nelle classi Individuare le funzioni da rendere virtuali

Reti e Protocolli	<u>Aspetti evolutivi delle reti</u> <u>Servizi per gli utenti e per le aziende</u> <u>Modello client/server</u> <u>Modello peer to peer</u> <u>La tecnologia di trasmissione</u> <u>Regole per il trasferimento dei dati</u> <u>Estensione delle reti</u> <u>Topologie di rete</u> <u>Tecniche di commutazione</u> <u>Architetture di rete</u> Livelli del modello ISO/OSI Mezzi trasmissivi Modello TCP/IP Indirizzi IP Classi di Indirizzi Indirizzi IPv6 Livelli applicativi nel modello TCP/IP Standard digitali per le reti pubbliche Tecnologia per la connettività mobile	Controllare la configurazione di rete del computer Individuare risorse condivise Descrivere le caratteristiche di una rete Individuare le unità che compongono una rete di computer Rappresentare con uno schema la topologia di una rete Rappresentare con uno schema a livelli un'attività di comunicazione Individuare gli aspetti rilevanti dei primi due livelli OSI Descrivere le caratteristiche tecniche dei dispositivi di rete Individuare gli standard utilizzati nei diversi ambiti Determinare l'indirizzo IP e la netmask di un computer Calcolare l'indirizzo della rete usando la netmask Scrivere esempi di indirizzi in formato IPv6 Descrivere le caratteristiche di una linea ADSL Individuare le differenze tra le tecnologie per la connettività mobile
Le basi di dati	Gestione automatizzata dei dati (DBMS) Progettazione di basi di dati: descrizione della realtà da modellare, modellazione della struttura dei dati a supporto di tale realtà, definizione dello schema della base dei dati, implementazione di un database mediante ACCESS ACCESS: tabelle, query, maschere, report	Conoscere gli ambiti di utilizzo dei DBMS Saper analizzare semplici realtà lavorative, produrre uno schema ER e costruire un database minimale utilizzando ACCESS
<b>QUINTE</b>		
Content and Language Integrated Learning (CLIL)	Computer architecture Codes and compression (HUFFMAN encoding)	Saper utilizzare il linguaggio specifico dell'informatica nell'ambito dell'architettura dei computer, dei codici e della compressione dei dati. In particolare saper descrivere la struttura interna di un computer e le relazioni tra le varie componenti in lingua inglese Saper esporre i concetti base della teoria dei codici Saper descrivere ed eseguire l'algoritmo di HUFFMAN
Programmazione ad oggetti (JAVA)	Astrazione Programmazione procedurale e strutturata Classi, attributi e metodi Incapsulamento, Information hiding Costruttore Accesso di tipo public e private alle member function Messaggi e interfaccia Ereditarietà Gerarchia delle classi Polimorfismo Membri protetti Overloading dei metodi Overriding dei metodi Overloading degli operatori delle classi Vantaggi della programmazione ad oggetti	Applicare i principi della programmazione ad oggetti utilizzando il C++ Creare una classe con attributi e metodi Dichiarare le istanze di una classe Definire un costruttore ed un distruttore di una classe Definire i costruttori Creare una classe derivata Saper utilizzare le clausole public e private Proteggere i membri di una classe Applicare l'overloading ai metodi delle classi derivate Saper riscrivere i metodi delle classi derivate Applicare l'overloading agli operatori nelle classi
Le basi di dati	Gestione automatizzata dei dati (DBMS) Progettazione di basi di dati: descrizione della realtà da modellare, modellazione della struttura dei dati a supporto di tale realtà, definizione dello schema della base dei dati, implementazione di un database mediante ACCESS ACCESS: tabelle, query, maschere, report	Conoscere gli ambiti di utilizzo dei DBMS Saper analizzare semplici realtà lavorative, produrre uno schema ER e costruire un database minimale utilizzando ACCESS
Crittografia	Crittografia antica: cifrario di Giulio Cesare, Vigenere Numeri primi Crittografia simmetrica e asimmetrica (a chiave pubblica): RSA	Conoscere l'importanza della Crittografia nella storia Conoscere l'algoritmo di codifica e decodifica RSA

## SPAZI

Laboratorio di Informatica, aula.

## METODI

La metodologia usata consiste in lezioni frontali interattive per l'individuazione di nessi, relazioni e concetti, adeguandosi di volta in volta alle esigenze della classe.

Particolare attenzione verrà prestata ad un progressivo arricchimento e ad un rigore del linguaggio specifico della materia.

L'unità didattica verrà presentata procedendo, ove possibile, attraverso l'intuizione comune della realtà, con l'individuazione di un problema, cercando poi le soluzioni con tecniche preesistenti e, se insufficienti, con l'introduzione di metodi e concetti nuovi arrivando quindi alla generalizzazione.

Verrà assegnato costantemente del lavoro da svolgere autonomamente, allo scopo di permettere allo studente una verifica personale dell'acquisizione delle conoscenze e della relativa corretta applicazione.

Seguirà in classe il confronto dei diversi procedimenti utilizzati e l'analisi degli eventuali errori compiuti oltre che, naturalmente, della/e soluzione/i.

L'uso degli strumenti informatici verrà introdotto in modo critico, senza creare l'illusione che esso sia un mezzo automatico di risoluzione di problemi e senza compromettere la necessaria acquisizione di calcolo mentale.

## MEZZI (manuali in adozione, LIM, dispense....)

-Testo in adozione:

**Classi prime e seconde:** Corso di Informatica per il nuovo liceo scientifico opzione scienze applicate, P. Camagni, R. Nikolassy, HOEPLI.

**Classi terze e quarte:** Informatica applicazioni scientifiche per il Liceo Scientifico delle Scienze applicate, A. Lorenzi, M. Govoni, ATLAS.

**Classi quinte:** Il linguaggio JAVA, Guida alla programmazione di base, M. Bigatti, HOEPLI;

Corso di Sistemi/Sistemi discreti, F. Cerri, HOEPLI.

In aggiunta, per tutte le classi: dispense fornite dall'insegnante.

## CRITERI DI VALUTAZIONE

Per tutte le classi la valutazione del raggiungimento degli obiettivi di apprendimento può avvenire mediante:

1. Prove non strutturate: interrogazione, compito scritto.
2. Prove strutturate: domande a risposta multipla, corrispondenze, completamento.
3. Prove semistrutturate: quesiti le cui risposte sono soggette a vincoli dettati dal contesto e/o dalla richiesta come quesiti a "risposta breve", problemi che richiedono procedure univoche di risoluzione.
4. Prove di laboratorio: creazione/scrittura/realizzazione di programmi in laboratorio.
5. Controlli e osservazioni: controllo dell'impegno dello studente relativamente a:
  - 5.1 correttezza dello svolgimento delle consegne domestiche;
  - 5.2 partecipazione attiva e attenzione durante le lezioni.

Al mancato rispetto delle consegne domestiche (non svolgimento dei compiti per casa) potranno corrispondere note disciplinari inserite nel registro.

La valutazione finale terrà conto, oltre che delle valutazioni conseguite nelle prove, dell'impegno, della costanza nello studio e della partecipazione durante tutta l'attività didattica.

Il Dipartimento ha concordato un numero minimo di verifiche da svolgere nelle modalità di cui al punto precedente senza alcun tipo di vincolo sulla tipologia, pari a:

-classi prime, seconde, terze e quarte: 2 (trimestre) + 2 (pentamestre);

-classi quinte: 3 (trimestre) + 3 (pentamestre).

Il docente ha la piena libertà di incrementare il numero di verifiche alla luce di particolari esigenze didattiche. I voti attribuibili alle prove appartengono all'intervallo 1-10, estremi inclusi.

Nella valutazione delle prove è fatto obbligo considerare il grado di conoscenza dello specifico argomento ovvero la conoscenza dei contenuti e delle regole, l'applicazione corretta degli algoritmi di calcolo, l'uso del linguaggio appropriato e della coerenza logica nonché la capacità di rielaborazione attraverso uno svolgimento ben organizzato e con ricerca del percorso ottimale di risoluzione.

Le griglie allegate permetteranno di esprimere la valutazione.

### Obiettivi

1. Capacità di interpretare la richiesta specifica centrando i punti nodali dell'argomento proposto (pertinenza);
2. Conoscenza dei contenuti;
3. Uso corretto di:
  - a) Terminologia specifica
  - b) Lingua a livello lessicale, morfologico, sintattico
4. Capacità espositiva sintetica/organica.

### GRIGLIA DI VALUTAZIONE DELLA PROVA SCRITTA/DI LABORATORIO/ORALE

Indicatori	Descrittori	Punti
Pertinenza della risposta	Inadeguata Adeguate Completa	1 2 3
Conoscenza dei contenuti	Scarsa / parziale Imprecisa Essenziale/Basilare Precisa Esauriente / Approfondita	1-2 3 4 5 6
Uso del linguaggio	Scorretto / improprio Parzialmente corretto Con imprecisioni Corretto/adequato	0 1 2 3
Capacità espositiva/organizzativa	Disorganica/incomprensibile Comprensibile Organica/efficace	1 2 3
	<b>Punteggio massimo</b>	<b>15</b>

### Livello di sufficienza

Indicatori + descrittori	Punti
Pertinenza adeguata	2
Conoscenza essenziale/basilare	4
Uso del linguaggio con imprecisioni	2
Capacità espositiva comprensibile	2
<b>Totale</b>	<b>10</b>

### Conversione in decimi

Quindicesimi	Decimi	Quindicesimi	Decimi	Quindicesimi	Decimi
1	1	6	3,8	11	6,8
2	1,6	7	4,3	12	7,6
3	2,1	8	4,9	13	8,4
4	2,7	9	5,4	14	9,2
5	3,2	10	6	15	10

### CRITERI E MODALITA' DI RECUPERO

Il dipartimento ritiene che le attività di recupero possano essere efficaci se tempestivamente attivate durante l'intero percorso didattico e in tal modo raccordate strettamente con lo svolgimento del programma.

In tal senso ritiene fondamentale il recupero in itinere, in quanto generalmente più tempestivo e continuativo degli eventuali interventi pomeridiani. Fanno parte integrante delle attività finalizzate al recupero in itinere:

a) le ore specificamente dedicate alla correzione e alla discussione degli esercizi e dei problemi assegnati per casa;

b) le esercitazioni in vista delle prove scritte;

c) le correzioni e discussioni dei compiti o dei test svolti in classe;

d) le interrogazioni viste come occasioni preziose per l'intera classe di riprendere, chiarire e riorganizzare i contenuti.

In caso di necessità potranno essere attivati corsi di recupero extracurricolari.

Vicenza , 3 Novembre 2014

IL DOCENTE  
Prof. Alessandro Peroni