

IIS "G. Cigna G. Baruffi F. Garelli"	Programmazione individuale Piano didattico annuale
--------------------------------------	---

Materia	Informatica
---------	-------------

Rif. Programmazione dipartimentale	<i>Segue la Progr. Del Dipartimento di "Matematica Applicata e Informatica "</i>
Triennio	Amministrazione Finanza e Marketing, Articolazione Sistemi Informativi Aziendali
Classe	Classe 4 ^A SIA

ACCORDO CON LA CLASSE

Durante le prime lezioni è stata illustrata la programmazione didattica annuale, per quanto concerne le tematiche, e la cadenza delle verifiche di profitto, nei due periodi didattici.

Si è inoltre sottolineata la necessità di prendere appunti su un quaderno, preferibilmente grande ad anelli (condivisibile in questo modo anche per altre materie, e evitando tablet e, PC, fogli volanti per questa attività). Più formalmente: si richiede agli allievi:

- ◆ partecipazione attiva durante le lezioni;
- ◆ dialogo costruttivo con l'insegnante e con i compagni;
- ◆ rispetto delle regole della convivenza scolastica;
- ◆ disponibilità all'ascolto e al rispetto reciproco;
- ◆ senso di responsabilità: conoscenza dei propri diritti e doveri.

PROGRAMMAZIONE INDIVIDUALE

La programmazione individuale, di seguito specificata in termini di Unità di Apprendimento, potrà, ovviamente essere variata in corso d'anno, a seconda del progredire della classe, di riduzione delle ore di lezione per impreviste chiusure della scuola, o per imprevedibili inaccessibilità e **difficoltà nel rendere adeguati i laboratori**. La programmazione di seguito pianificata prevede per ogni UDA mirate e graduali esercitazioni in laboratorio concordate con il docente di laboratorio.

UNITÀ DI APPRENDIMENTO 1 : Programmazione Python e Concetti avanzati di Object Oriented

<p>COMPETENZE:</p> <p>Modellare semplici processi aziendali tramite il paradigma OO</p> <p>Comprendere e Estendere applicativi per le proprie esigenze</p> <p>MACRO CONOSCENZE paradigma object oriented Linguaggi di programmazione Metodologia di sviluppo di software Fasi di sviluppo di un progetto software</p>	<p>OBIETTIVI SPECIFICI</p> <ul style="list-style-type: none"> ■ Esprimere procedimenti risolutivi attraverso algoritmi ■ Implementare algoritmi con diversi stili di programmazione e idonei strumenti software ■ Produrre la documentazione relativa alle fasi di progetto ■ Valutare, scegliere e <u>adattare</u> software applicativi in relazione alle caratteristiche e al fabbisogno aziendale 	<p>PERIODO: Settembre/ Gennaio (60 Ore~)</p>
<p>CONTENUTO:</p> <ul style="list-style-type: none"> • Ambiente di sviluppo Integrato, presenti in laboratorio • Il linguaggio Python e l'interprete (storia, motivazioni, evoluzione) • The Python Standard Library , Documentazione Help online • Caratteristiche del linguaggio: • Operatori aritmetici, di confronto, logici (connettivi in e not).. • Variabili, identificatori e tipi del linguaggio (linguaggio tipato dinamicamente → non necessaria dichiarazione di tipo, la funzione type()) • L'output e l'input (le funzioni built-in input() print() , l'output formattato e le "f strings"). • il controllo del flusso condizionale (if – else , elif) • Le collezioni: Liste, Tuple, Set, Dizionari • Le stringhe in python • I cicli while e for (ciclo for "for each") break continue else • procedure e funzioni • Il paradigma Object Oriented differenze rispetto all'approccio imperativo (analisi dei vari paradigmi di programmazione) • I Concetti di Classe oggetto, attributi e metodi • Linguaggio UML <ul style="list-style-type: none"> → Cenni al linguaggio UML → Il diagramma delle Classi • I Principi di <i>Data Hiding</i> e di <i>Incapsulamento</i> (la visibilità nel paradigma Object oriented), l'interfaccia • il costruttore • L'esecuzione di un programma concepito con la metodologia OO: lo scambio di messaggi tra oggetti e la dot notation • Concetti Avanzati del paradigma OO: <ul style="list-style-type: none"> → Relazione IS-A ed <i>ereditarietà</i> : Superclasse Sottoclasse e meccanismo dell'ereditarietà (notazione UML sul diagramma delle classi). Ereditarietà Singola Vs. Ereditarietà Multipla → Motivazione all'adozione dell'ereditarietà → Evoluzione della Visibilità dei membri di una classe nella relazione IS-A , lo specificatore <i>protected</i> → design pattern <i>cenni</i> → L'ereditarietà con ridefinizione dei metodi – il polimorfismo degli oggetti: overriding e overloading Altre relazioni tra classi differenti da quella IS-A (composizione aggregazione) <p>SPERIMENTAZIONI</p> <ul style="list-style-type: none"> • I costrutti offerti dal linguaggio Python per supportare il paradigma Object Oriented • La definizione di una classe (convenzioni sulla nomenclatura dell'identificatore di una classe). <i>class</i>. • La "dichiarazione" dei membri della classe <ul style="list-style-type: none"> → variabili di Istanza (approfondimento: differenza con le variabili locali, inizializzazione e visibilità) → metodi di Istanza la firma di un metodo → il metodo costruttore • specificatore di accesso ai membri di una classe public e private • Ereditarietà in python • la valutazione di un'espressione di assegnazione in presenza di ereditarietà • la valutazione di una chiamata di metodo in presenza di estensione e ridefinizione 		<p>TIPOLOGIA DI VERIFICA</p> <ul style="list-style-type: none"> • Verifica Scritta anche con domande aperte • Test a risposta multipla • Eventuale interrogazione • Produzione di elaborati (programmi)
		<p>METODOLOGIA:</p> <ul style="list-style-type: none"> • Lezione frontale partecipata (esercizi o interventi alla lavagna) • Uso della LIM, videoriproduzione slide docente • Uso del libro di testo (rif. Unità 1,2) • <i>Manualistica Python di riferimento condivisa con la classe</i>

<p>(binding dinamico)</p> <ul style="list-style-type: none">• keyword this e super (in presenza di ereditarietà invocazione membri superclasse..)• File e operazioni per la persistenza• Le collezioni in Python (oltre l'array, strutture dti dinamiche)•• Si illustrano/<u>ripassano</u> alcune scorciatoie di uso frequente nell IDE pycharm CE• esercitazione volte a implementare applicazioni semplificate oo: gestione di una rubrica, di conti in una contabilità semplificata, di libri (biblioteca), di dipendenti..	
--	--

UNITÀ DI APPRENDIMENTO 2 : - DBMS Relazionali

<p>COMPETENZE: Modellare la realtà del dominio di interesse per gestire i dati con DBMS relazionali Saper interagire attraverso linguaggio SQL con DBMS esistenti</p> <p>MACRO CONOSCENZE</p> <ul style="list-style-type: none"> • Data Base Management System (DBMS) • Progettazione di Data Base • Linguaggio SQL • Data base in rete 	<p>OBIETTIVI SPECIFICI</p> <ul style="list-style-type: none"> ■ Progettare e realizzare basi di dati in relazione alle esigenze aziendali ■ Operare con DBMS relazionali esistenti (anche attraverso linguaggi di programmazione) 	<p>PERIODO: Febbraio- Marzo Aprile 60h</p>
<p>CONTENUTO:</p> <ul style="list-style-type: none"> • Introduzione ai DBMS nel contesto del sistema informativo aziendale (Motivazioni alla loro introduzione visto il fallimento di approcci alternativi) • L'architettura di un DBMS e i modelli usati per descrivere i dati a differenti livelli. • Progettazione di una Base di dati Relazionale: <ul style="list-style-type: none"> → Modellazione Concettuale (Modello ER) → il Modello Relazionale dei dati → Da modello concettuale ER a modello logico Relazionale → Forme normali • Utilizzo di un DBMS Relazionale <ul style="list-style-type: none"> → algebra relazionale (operazioni proiezione, selezione, join e loro applicazione) → operatori insiemistici (Unione, differenza, intersezione, prodotto cartesiano e loro applicazione) → il linguaggio SQL <ul style="list-style-type: none"> ◆ DDL (tipi di dati, creazione domini, comandi per la dichiarazione delle tabelle, vincoli di integrità referenziale, cancellazione e modifica) ◆ DML ,(operatori aggiornamento inserzione, modifica cancellazione) DQL (gli operatori algebrici in SQL:selezione, proiezione, join e varianti, IN, LIKE, ORDER BY, SUM, AVG, MAX, MIN, COUNT, GROUP BY e HAVING) <p>SPERIMENTAZIONI:</p> <ul style="list-style-type: none"> • La gestione della persistenza degli applicativi, FILE. Gestione con Record a lunghezza variabile e fissa. La persistenza in Python Esercitazioni. • SQL, DML e DQL : Applicazione degli operatori dell'algebra relazionale e insiemistici gradualmente a modelli di una sola relazione e poi alcuni casi con più relazioni. Interagendo da linea di comando verso DBMS, o inserendo le query in codice Python • SQL DDL Creazione dello schema logico progettato • 		<p>TIPOLOGIA DI VERIFICA</p> <ul style="list-style-type: none"> • Verifica Scritta anche con domande aperte (esercizi di progettazione, o creazione di query) • Test a risposta multipla • interrogazione • <p>METODOLOGIA:</p> <ul style="list-style-type: none"> • Lezione frontale partecipata (esercizi o interventi alla lavagna) • Uso della LIM, videoriproduzione slide docente • Sperimentazione costante • Uso del libro di testo (rif. Unità 5,6,7,8,9, 10) •

UNITÀ DI APPRENDIMENTO 3 : - Sviluppo di Applicazioni multi tier “web”

COMPETENZE: Creazione di pagine semplici pagine web statiche Realizzazione di Servizi web a supporto di comune esigenze dell'organizzazione aziendale MACRO CONOSCENZE Progettazione d'ipermedia per la comunicazione aziendale Linguaggi e strumenti di implementazione per il Web Struttura, usabilità e accessibilità di un sito Web Reti di computer e reti di comunicazione Data base in rete Servizi di rete a supporto dell'azienda E-commerce Social networking	OBIETTIVI SPECIFICI <ul style="list-style-type: none">■ Individuare le procedure telematiche che supportano l'organizzazione di un'azienda■ Implementare data base remoti con interfaccia grafica sul web in relazione alle esigenze aziendali■ Progettare ipermedia a supporto della comunicazione aziendale■ Progettare e realizzare pagine Web statiche e dinamiche■ Pubblicare su Internet pagine Web	PERIODO: Aprile MAggio 32h
CONTENUTO: <ul style="list-style-type: none">• Comunicazione tra Calcolatori, il paradigma client/server e il protocollo applicativo HTTP• Linguaggi del web (HTML, CSS, XML)• Sviluppo di Applicazioni multi-tier con business logic (Application server), DBMS, e livello di presentazione• SPERIMENTAZIONI <ul style="list-style-type: none">• Pagine web statiche HTML CSS (eventualmente js e DOM)• Realizzazione di applicazioni con tecnologie PHP/Python che facciano uso di dati su DBMS	TIPOLOGIA DI VERIFICA <ul style="list-style-type: none">• Verifica Scritta anche con domande aperte (Progettazione di pagine statiche, o dinamiche con le tecnologie illustrate). Anche al calcolatore.• Test a risposta multipla• interrogazione METODOLOGIA: <ul style="list-style-type: none">• Lezione frontale partecipata (esercizi o interventi alla lavagna)• Uso della LIM, videoriproduzione slide docente• Sperimentazione costante• Uso del libro di testo (<i>rif. Unità 3, 4</i>) <i>Documentazione php.net</i> <i>Documentazione W3c</i>	