

# PROGRAMMAZIONE DIDATTICA

Disciplina: **INFORMATICA**

Classe: 3<sup>^</sup> *I.T.I.S corso Abacus*

*Anno Scolastico 2009/2010*

## **Finalità:**

Il corso di Informatica ha come fine principale quello di mettere in grado lo studente di affrontare la soluzione di un problema posto dalla richiesta di un ipotetico committente, scegliendo le metodologie e gli strumenti software più idonei, e di fornirgli inoltre la formazione di base che gli consenta di seguire con una certa autonomia l'evoluzione delle tecnologie informatiche.

## **Obiettivi:**

L'obiettivo della programmazione curricolare annuale è introdurre l'alunno alla programmazione evidenziando l'esistenza di diversi modi di intendere la programmazione stessa ed abituarlo a ragionare per modelli. Al termine della terza classe gli allievi dovranno raggiungere i seguenti obiettivi:

- Analizzare criticamente le conoscenze pregresse di informatica.
- Impostare problemi, anche da un punto di vista non procedurale.
- Risolvere problemi, indipendentemente da un linguaggio di programmazione.
- Verificare la correttezza di una soluzione.
- Leggere ed interpretare descrizioni sintattiche in più notazioni.
- Applicare corretti stili di programmazione.
- Documentare il software a livello elementare.
- Usare con proprietà un linguaggio imperativo.
- Usare il linguaggio di comando di un sistema operativo.

Tali conoscenze permetteranno agli allievi di affrontare nelle due classi successive argomenti sempre più complessi.

## **Contenuti:**

### Introduzione globale alle discipline informatiche:

- Concetto e ruoli dell'informazione.
- Origini matematiche e tecnologiche dell'informatica.
- Ramificazioni principali dell'informatica.
- Ambienti e figure professionali.
- Reti di computer e rete Internet.
- Previsioni tecnologiche e di mercato a breve e medio termine.

### Problemi e programmi:

- Definizione di problema e dati; strategie di soluzione e definizione di algoritmo.
- Analisi e definizione dei dati essenziali, superflui, di Input, Output, variabili, costanti.
- Le strutture di controllo del flusso dei dati.
- Formalizzazione degli algoritmi mediante linguaggio di progetto e/o diagramma a blocchi.

### Elementi di sintassi dei linguaggi:

- Simbolo, alfabeto, stringa, linguaggio.
- Linguaggi artificiali.
- Concetti di riconoscimento e di generazione di un linguaggio.
- Interpreti e compilatori.

### Introduzione alla programmazione imperativa: dall'analisi all'esecuzione pratica:

- L'ambiente di programmazione Dev-C++.
- Strutture di controllo e programmazione strutturata.
- Variabili, espressioni, assegnazione, tipi elementari.
- Funzioni e procedure.
- Passaggio dei parametri e regole di visibilità.
- Programmazione ricorsiva.

### Dati strutturati:

- Organizzazione dati mediante la struttura vettoriale.
- Metodi di ricerca in un vettore
- Ordinamento di un vettore.
- Organizzazione dei dati mediante matrici.
- Dati complessi: il record.
- Organizzazione dei dati tramite vettori di record.

### Tecniche di produzione del software:

- Fattori di qualità del SW.
- Metodologie di sviluppo top-down e bottom-up.
- Documentazione del SW: tecniche elementari.

### Elementi di programmazione non procedurale:

- Introduzione al paradigma logico e funzionale.
- Attività di confronto tra i paradigmi.

### La comunicazione con il computer:

- I servizi di Internet.
- Il linguaggio HTML e le pagine per i siti Internet.
- I fogli di stile a cascata.
- Cenni su usabilità ed accessibilità dei siti web.

### **Modalità di lavoro:**

1. Spiegazioni in classe e/o in laboratorio.
2. Utilizzo del laboratorio di Informatica.
3. Studio e svolgimento di esercizi a casa.
4. Assegnazione di compiti con stesura di relazioni.
5. Recupero periodico.
6. Attività di approfondimento e ricerca.

### **Strumenti di lavoro:**

1. Lavagna.
2. Proiettore di lucidi.
3. Testi in adozione.
4. Manuali di programmazione.
5. Laboratorio di Informatica.
6. Fotocopie di materiale didattico vario (articoli di riviste, appunti, schemi, ecc.).

### **Tipologie e numero di verifiche:**

1. Verifiche formative orali, scritte, pratiche (se necessarie, su singole lezioni ed argomenti).
2. Verifiche sommative orali, scritte, pratiche.
3. Verifiche di recupero orali, scritte, pratiche (se necessarie).
4. Relazioni scritte in laboratorio o a casa

### **Numero di verifiche:**

Per ogni quadrimestre ogni studente sarà sottoposto mediamente a quattro prove di verifica sommativa di tipologia varia.

### **Requisiti minimi per l'attribuzione della sufficienza:**

Lo studente:

- possiede i contenuti minimi: teoria dell'informazione, basi dei linguaggi di programmazione, definizione di algoritmo, strutture di controllo, dati strutturati, funzioni e procedure ,basi di C, elementi di html e css
- applica le conoscenze acquisite per la soluzione di problemi noti (saper analizzare e risolvere semplici problemi indipendentemente dal linguaggio di programmazione, saper sviluppare semplici programmi in C , realizzare semplici pagine in html)
- si esprime in modo semplice ma con uso del linguaggio specifico della materia, cerca soluzioni anche precostituite.

Disciplina: **Informatica**

PIANO DI LAVORO Classe: 3<sup>a</sup> I.T.I.S. corso Abacus

Anno Scolastico 2009/2010

PERIODO	ARGOMENTI	CONTENUTI	OBIETTIVI	MODALITÀ	STRUMENTI	VERIFICHE	ORE
Settembre Ottobre Novembre	La disciplina informatica e il sistema di elaborazione dati Le basi della programmazione Gli algoritmi Le strutture di controllo Gli strumenti di sviluppo software	<ul style="list-style-type: none"> <li>- Introduzione globale alle discipline informatiche.</li> <li>- Problemi e programmi.</li> <li>- Elementi di sintassi dei linguaggi.</li> <li>- Introduzione programmazione imperativa.</li> <li>- Fondamenti dell'ambiente di sviluppo C</li> </ul>	<ul style="list-style-type: none"> <li>- Acquisizione degli strumenti e delle capacità di base per l'analisi e la soluzione di un problema.</li> <li>- Sapere risolvere semplici problemi indipendentemente dal linguaggio di programmazione</li> <li>- Saper sviluppare programmi elementari in C.</li> </ul>	1 - 2 - 3 - 5 - 6	1 - 2 - 3 - 4 - 5 - 6	1 - 2 - 3 - 4	65
Dicembre Gennaio Febbraio Marzo	Il linguaggio delle pagine web Tecniche evolute di programmazione	<ul style="list-style-type: none"> <li>- Introduzione al linguaggio HTML e i CSS.</li> <li>- Programmazione imperativa.</li> <li>- Dati strutturati</li> <li>- Tecniche di produzione del software.</li> </ul>	<ul style="list-style-type: none"> <li>- Saper sviluppare semplici pagine web.</li> <li>- Conoscenza approfondita del linguaggio imperativo C.</li> <li>- Saper utilizzare dati strutturati.</li> <li>- Saper applicare corretti stili di programmazione.</li> </ul>	1 - 2 - 3 - 4 - 5 - 6	1 - 2 - 3 - 4 - 5 - 6	1 - 2 - 3 - 4	75
Aprile Maggio Giugno	La multimedialità e le reti	<ul style="list-style-type: none"> <li>- La comunicazione con il computer</li> </ul>	<ul style="list-style-type: none"> <li>- Conoscere approfonditamente il linguaggio HTML e i CSS</li> </ul>	1 - 2 - 3 - 4 - 5 - 6	1 - 2 - 3 - 4 - 5 - 6	1 - 2 - 3 - 4	55

Disciplina: **Informatica – Esercitazioni in laboratorio**

PIANO DI LAVORO  *Classe: 3<sup>^</sup> I.T.I.S. corso Abacus*

Anno Scolastico 2009/2010

PERIODO	TEMA ESERCITAZIONI	OBIETTIVI	MODALITÀ	STRUMENTI	VERIFICHE	ORE
Settembre Ottobre Novembre	<ol style="list-style-type: none"> <li>1. Ambiente Dev-C++.</li> <li>2. Sviluppo programmi per operazioni di ingresso e uscita dati</li> <li>3. Sviluppo programmi con struttura sequenziale, alternativa, iterativa</li> <li>4. Sviluppo programmi con utilizzo di stringhe.</li> </ol>	<ul style="list-style-type: none"> <li>– Acquisizione degli strumenti e delle capacità di base per l'analisi e la soluzione di un problema.</li> <li>– Saper sviluppare programmi in C elementari.</li> <li>– Saper applicare corretti stili di programmazione (uso di indentazione e commenti).</li> </ul>	1-2-3-4-5-6	1-2-3-4-5-6	1-2-3-4	33
Dicembre Gennaio Febbraio Marzo	<ol style="list-style-type: none"> <li>5. Esercitazioni relative ai primi tag HTML.</li> <li>6. Sviluppo programmi con funzioni senza parametri.</li> <li>7. Sviluppo programmi con uso dei vettori</li> <li>8. Sviluppo programmi con funzioni con parametri.</li> <li>9. Sviluppo programmi con uso di matrici.</li> <li>10. Ordinamento di vettori, vettori paralleli.</li> <li>11. Utilizzo di strutture di dati.</li> </ol>	<ul style="list-style-type: none"> <li>– Saper sviluppare semplici pagine web.</li> <li>– Conoscenza approfondita del linguaggio imperativo C.</li> <li>– Utilizzare dati strutturati e complessi</li> </ul>	1-2-3-4-5-6	1-2-3-4-5-6	1-2-3-4	38
Aprile Maggio Giugno	<ol style="list-style-type: none"> <li>12. Problemi ricorsivi.</li> <li>13. Sviluppo di applicativi completi con menu di scelta</li> <li>14. Realizzazione di un sito web completo</li> </ol>	<ul style="list-style-type: none"> <li>– Conoscenza approfondita del linguaggio imperativo C.</li> <li>– Conoscere approfonditamente il linguaggio HTML e i CSS</li> </ul>	1-2-3-4-5-6	1-2-3-4-5-6	1-2-3-4	28

## GRIGLIA DI DEFINIZIONE DEI REQUISITI MINIMI PER L'ATTRIBUZIONE DELLA SUFFICIENZA

<b>MATERIA:</b> informatica	<b>CLASSE:</b> 3 <sup>^</sup>	<b>INDIRIZZO:</b> informatica Abacus
-----------------------------	-------------------------------	--------------------------------------

<i>CONTENUTI</i>	<i>CONOSCENZE MINIME PER L'ACCESSO ALLA CLASSE SUCCESSIVA</i>	<i>COMPETENZE MINIME PER L'ACCESSO ALLA CLASSE SUCCESSIVA</i>
<ul style="list-style-type: none"> <li>• <i>Basi dei linguaggi di programmazione</i></li> <li>• <i>Definizione di algoritmo</i></li> </ul>	<ul style="list-style-type: none"> <li>• Saper analizzare e risolvere semplici problemi indipendentemente dal linguaggio di programmazione</li> </ul>	<ul style="list-style-type: none"> <li>• Applica le conoscenze acquisite per la soluzione di problemi noti</li> <li>• Si esprime in modo semplice ma con uso del linguaggio specifico della materia, cerca soluzioni anche precostituite</li> </ul>
<ul style="list-style-type: none"> <li>• <i>Strutture di controllo</i></li> <li>• <i>Dati elementari e strutturati</i></li> </ul>	<ul style="list-style-type: none"> <li>• Saper utilizzare le strutture di controllo</li> <li>• Saper utilizzare tipi di dati primitivi</li> <li>• Saper utilizzare vettori e matrici</li> </ul>	
<ul style="list-style-type: none"> <li>• <i>Funzioni e procedure</i></li> <li>• <i>Passaggio di parametri</i></li> </ul>	<ul style="list-style-type: none"> <li>• Saper suddividere i problemi in sotto-problemi con l'uso di funzioni e procedure</li> <li>• Saper utilizzare le basi dell'uso dei parametri passati per copia o per riferimento</li> </ul>	
<ul style="list-style-type: none"> <li>• <i>Elementi di C</i></li> </ul>	<ul style="list-style-type: none"> <li>• Saper sviluppare semplici programmi in C che utilizzano dati elementari, dati strutturati e semplici funzioni</li> </ul>	
<ul style="list-style-type: none"> <li>• <i>Elementi di Html e Css</i></li> </ul>	<ul style="list-style-type: none"> <li>• Saper realizzare semplici pagine in Html con utilizzo di elementi base dei Css</li> </ul>	

# PROGRAMMAZIONE DIDATTICA

Disciplina: **INFORMATICA**

Classe: 4<sup>^</sup> *I.T.I.S corso Abacus*

*Anno Scolastico 2009/10*

## **Finalità:**

Il corso di Informatica ha come fine principale quello di mettere in grado lo studente di affrontare la soluzione di un problema, posto dalla richiesta di un ipotetico committente, scegliendo le metodologie e gli strumenti SW più idonei, ed inoltre fornirgli la formazione di base che gli consenta di seguire con una certa autonomia l'evoluzione delle tecnologie informatiche.

## **Obiettivi:**

L'obiettivo della programmazione curricolare annuale è far acquisire all'alunno le competenze per affrontare progetti di una certa dimensione anche con la metodologia orientata agli oggetti. Deve acquisire una certa flessibilità che gli consenta di analizzare e risolvere problemi di varia natura utilizzando il paradigma e gli strumenti più idonei alla natura del problema.

Al termine della quarta classe gli allievi dovranno raggiungere i seguenti obiettivi:

- Svolgere un'analisi qualitativa degli algoritmi.
- Applicare con proprietà tecniche di programmazione modulare.
- Elaborare file ad organizzazione sequenziale e random.
- Organizzazioni con archivi tradizionali - File sequenziali paginati - File hash - File con indici dinamici ad albero
- Definire e manipolare strutture di dati connesse da puntatori.
- Applicare il paradigma della programmazione orientata agli oggetti.
- Applicare il paradigma della programmazione guidata degli eventi.
- Progettare e costruire interfacce d'utente amichevoli.
- Conoscere ed usare classi notevoli di oggetti contenitori.
- Conoscere le basi della compilazione/interpretazione.
- Analizzare criticamente le conoscenze pregresse d'informatica.

Tali conoscenze permetteranno agli allievi di affrontare nella classe finale argomenti sempre più complessi.

## **Contenuti:**

### Ripasso e approfondimenti sulla programmazione imperativa

- Strutture dati complesse: array, matrici, record
- Algoritmi di ordinamento. Problemi ricorsivi
- File

### La comunicazione con il computer:

- Il linguaggio HTML e i CSS
- Il linguaggio JavaScript
- Uso di un applicativo per la creazione e la gestione di siti web

### L'analisi degli algoritmi:

- Analisi qualitativa degli algoritmi
- Valutazione della complessità computazionale

### Archivi e file:

- Definizione di archivio
- Concetti di record e chiave
- Operazioni sugli archivi
- Archivi sequenziali
- Archivi non sequenziali

### Allocazione dinamica della memoria:

- Puntatori e reference
- Gestione stack e heap

### La programmazione ad oggetti:

- Classi ed oggetti.
- Variabili locali e variabili istanza
- Attributi e metodi.

- Variabili e metodi di classe
- Interfacce con l'esterno, incapsulamento e information hiding
- Ereditarietà.
- Classi astratte.
- Gerarchie di oggetti
- Polimorfismo.
- Casting

#### Il linguaggio Java:

- Introduzione al linguaggio e all'ambiente di sviluppo Eclipse
- Programmazione ad oggetti
- Applet
- La grafica e le interfacce utente grafiche
- La gestione degli eventi

#### Strutture di dati:

- Introduzione alle strutture dati complesse
- Alcuni esempi di ADT (pila, coda, lista)
- Strutture dati non lineari (grafi e alberi)
- Gestione degli archivi ad oggetti

#### Fondamenti di ingegneria del software:

- Introduzione all'ingegneria del software
- Ciclo di vita del software
- Analisi e progettazione orientata agli oggetti.

#### **Modalità di lavoro:**

1. Spiegazioni in classe e/o in laboratorio.
2. Utilizzo del laboratorio di Informatica.
3. Studio e svolgimento di esercizi a casa.
4. Assegnazione di compiti con stesura di relazioni sotto forma di siti web.
5. Recupero periodico.
6. Attività di approfondimento e ricerca.

#### **Strumenti di lavoro:**

1. Lavagna.
2. Testi in adozione.
3. Manuali di programmazione.
4. Laboratorio di Informatica.
5. Fotocopie di materiale didattico vario (articoli di riviste, appunti, schemi, ecc.).

#### **Tipologie di verifica:**

1. Verifiche formative orali, scritte, pratiche (se necessarie, su singole lezioni ed argomenti).
2. Verifiche sommative orali, scritte, pratiche.
3. Verifiche di recupero orali, scritte, pratiche (se necessarie).
4. Relazioni scritte in laboratorio o a casa.

#### **Numero di verifiche:**

Per ogni quadrimestre ogni studente sarà sottoposto mediamente a quattro prove di verifica sommativa di tipologia varia.

#### **Requisiti minimi per l'attribuzione della sufficienza:**

Lo studente:

- possiede i contenuti minimi: strutture dati, file, ADT, programmazione a oggetti, basi del linguaggio Java, basi dell'ingegneria del sw
- applica le conoscenze acquisite per la soluzione di problemi noti (saper sviluppare programmi in C con uso dei file e di strutture dati, saper realizzare semplici programmi ad oggetti in Java, saper sviluppare interfacce grafiche in Java)
- si esprime in modo semplice ma con uso del linguaggio specifico della materia, cerca soluzioni anche precostituite.

PERIODO	ARGOMENTI	CONTENUTI	OBIETTIVI	MODALITÀ	STRUMENTI	VERIFICHE	ORE
Settembre Ottobre Novembre	Ripasso e approfondimenti elementi programmazione imperativa  Analisi degli algoritmi.  Progettazione siti web.  Gestione di archivi.	- Strutture dati complesse. - Algoritmi ricorsivi. - Archivi.  - Analisi qualitativa degli algoritmi. - La comunicazione con il computer. - Files sequenziali	- Utilizzare C per problemi ricorsivi e strutture dati complesse - Conoscere le definizioni di archivio, record, campo, chiave, organizzazione ed accesso ai dati. - Valutare la complessità computazionale. - Progettare e costruire siti web utilizzando HTML, CSS, JavaScript e uno specifico applicativo. - Elaborare file ad organizzazione ed accesso sequenziale.	1-2-3-4-5-6	1-2-5	1-2-3-4	65
Dicembre Gennaio Febbraio	Gestione di archivi.  Gestione di pile, code e liste.  Introduzione alla OOP (=programmazione Object-Oriented) ed al linguaggio Java.	- Files relativi  - Allocazione dinamica della memoria.  - La programmazione ad oggetti. - Introduzione al linguaggio Java	- Elaborare file ad organizzazione realtiva ed accesso sequenziale/random.  - Definire e manipolare strutture dati dinamiche. - Conoscere le tecniche di gestione dinamica della memoria. - Applicare il paradigma della programmazione orientata agli oggetti a problemi di varia natura.	1-2-3-5-6	1-2-5	1-2-3-4	65
Marzo Aprile Maggio Giugno	OOP: ereditarietà e polimorfismo. Java: le eccezioni, le interfacce grafiche e gli eventi. Ingegneria del software.	- Approfondimento del linguaggio Java  - Ingegneria del software	- Conoscere, progettare ed implementare algoritmi complessi con Java. - Progettare e costruire interfacce d'utente amichevoli. - Conoscere il ciclo di vita del SW e adottare corretti stili di programmazione.	1-2-3-5-6	1-2-5	1-2-3-4	65

PERIODO	TEMA ESERCITAZIONI	OBIETTIVI	MODALITÀ	STRUMENTI	VERIFICHE	ORE
Settembre Ottobre Novembre	<ol style="list-style-type: none"> <li>1. Sviluppo di problemi ricorsivi in C.</li> <li>2. Sviluppo di applicativi completi in C con menu di scelta, funzioni, strutture dati complesse.</li> <li>3. Sviluppo di pagine web con HTML e CSS</li> <li>4. Sviluppo di script in JavaScript</li> <li>5. Sviluppo di applicativi in C con l'uso di file.</li> </ol>	<ul style="list-style-type: none"> <li>– Saper utilizzare C per problemi ricorsivi e strutture dati complesse</li> <li>– Saper progettare e costruire siti web utilizzando HTML, CSS, JavaScript.</li> <li>– Saper utilizzare C per la gestione degli archivi scon accesso equenziale.</li> </ul>	1-2-3-4-5-6	1-2-3-4-5	1-2-3-4	33
Dicembre Gennaio Febbraio	<ol style="list-style-type: none"> <li>6. Sviluppo di applicativi in C con l'uso di file.</li> <li>7. Sviluppo semplici problemi di utilizzo dei puntatori con C( code, stack, liste).</li> <li>8. L'ambiente di sviluppo Eclipse.</li> <li>9. Sviluppo di semplici programmi Java con gerarchie di classi.</li> <li>10. Sviluppo di programmi Java con gestione delle eccezioni.</li> </ol>	<ul style="list-style-type: none"> <li>– Saper utilizzare C per la gestione degli archivi con accesso random.</li> <li>– Saper definire e manipolare strutture dati dinamiche.</li> <li>– Saper applicare il paradigma della programmazione orientata agli oggetti a problemi di varia natura.</li> </ul>	1-2-3-4-5-6	1-2-3-4-5	1-2-3-4	33
Marzo Aprile Maggio Giugno	<ol style="list-style-type: none"> <li>11. Sviluppo di programmi Java con interfaccia grafica.</li> <li>12. Sviluppo di programmi Java con uso degli stream.</li> <li>13. Applicativi completi di riepilogo/recupero</li> </ol>	<ul style="list-style-type: none"> <li>– Conoscere, progettare ed implementare strutture dati complesse con Java.</li> <li>– Saper elaborare file ad organizzazione sequenziale.</li> <li>– Progettare e costruire interfacce d'utente amichevoli.</li> </ul>	1-2-3-4-5-6	1-2-3-4-5	1-2-3-4	33

## GRIGLIA DI DEFINIZIONE DEI REQUISITI MINIMI PER L'ATTRIBUZIONE DELLA SUFFICIENZA

<b>MATERIA:</b> informatica	<b>CLASSE:</b> 4 <sup>^</sup>	<b>INDIRIZZO:</b> informatica Abacus
-----------------------------	-------------------------------	--------------------------------------

<i>CONTENUTI</i>	– <i>CONOSCENZE MINIME</i> – <i>PER L'ACCESSO ALLA CLASSE SUCCESSIVA</i>	<i>COMPETENZE MINIME</i> <i>PER L'ACCESSO ALLA CLASSE SUCCESSIVA</i>
<ul style="list-style-type: none"> <li>• <i>Dati strutturati</i></li> <li>• <i>File</i></li> <li>• <i>ADT</i></li> </ul>	<ul style="list-style-type: none"> <li>• Conoscere l'uso di record e tabelle anche in C</li> <li>• Conoscere le basi dell'archiviazione dati su memoria di massa</li> <li>• Conoscere l'uso di file e di dati strutturati in C</li> <li>• Conoscere alcuni tipi di dati astratti</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Applica le conoscenze acquisite per la soluzione di problemi noti</b></li> <li>• <b>Si esprime in modo semplice ma con uso del linguaggio specifico della materia, cerca soluzioni anche precostituite</b></li> </ul>
<ul style="list-style-type: none"> <li>• <i>Programmazione a oggetti</i></li> <li>• <i>Basi linguaggio Java</i></li> </ul>	<ul style="list-style-type: none"> <li>• Conoscere le basi della programmazione orientata agli oggetti</li> <li>• Conoscere le basi dell'linguaggio Java</li> <li>• Conoscere gli elementi di base per la costruzione di semplici interfacce grafiche in Java</li> </ul>	
<ul style="list-style-type: none"> <li>• <i>Basi dell'ingegneria del software</i></li> </ul>	<ul style="list-style-type: none"> <li>• Conoscere le basi dell'ingegneria del software</li> </ul>	

# PROGRAMMAZIONE DIDATTICA

Disciplina: **INFORMATICA**

Classe: 5<sup>a</sup> **I.T.I.S corso Abacus**

Anno Scolastico 2009/10

## **Finalità:**

Il corso di Informatica ha come fine principale quello di mettere in grado lo studente di affrontare la soluzione di un problema, posto dalla richiesta di un ipotetico committente, scegliendo le metodologie e gli strumenti software più idonei, ed inoltre fornirgli la formazione di base che gli consenta di seguire con una certa autonomia l'evoluzione delle tecnologie informatiche.

## **Obiettivi:**

L'obiettivo della programmazione curricolare annuale è far acquisire all'alunno le competenze necessarie per rappresentare e gestire un insieme di informazioni, nonché per saper scegliere il tipo di organizzazione più adatto a seconda dell'applicazione. L'alunno dovrà saper gestire il progetto e la manutenzione di Sistemi Informativi per piccole realtà e conoscere i concetti e le tecniche fondamentali per la progettazione di basi di dati.

Al termine della quinta classe gli allievi dovranno raggiungere i seguenti obiettivi:

- Utilizzare con sicurezza un linguaggio general-purpose quale Java.
- Gestione di informazioni. Il progetto dei sistemi informativi. Modellazione concettuale di un sistema informatico (ad es. con il modello entità-associazioni).
- Basi di dati Concetti generali sulle basi di dati.
- Sistemi per la Gestione di Basi di Dati (DBMS).
- Il Modello relazionale e sue operazioni.
- Linguaggi di interrogazione non procedurali (SQL).
- Problematiche in multiutenza. Tecniche di accesso ai database in rete. Programmazione lato sever.
- La produzione del software. Strumenti software di supporto allo sviluppo del software. Documentazione e presentazione del software.

## **Contenuti:**

### Ripasso ed approfondimenti sulla programmazione a oggetti ed il linguaggio Java:

- Gestione delle eccezioni.
- Gestione dei files sequenziali e random.

### Archivi, basi di dati e DBMS:

- Il sistema informativo aziendale. I sistemi informativi e la loro gestione automatica.
- Analisi dei problemi con archiviazione dati. Rappresentazione di situazioni reali e di processi aziendali.
- Caratteristiche del database. Vantaggi rispetto alle tecniche tradizionali.
- Modelli logici di rappresentazione dati: gerarchico, reticolare, relazionale.
- Modello relazionale e sue caratteristiche.
- Tecniche di analisi: livello concettuale → livello logico → livello fisico.
- Modello concettuale Entità-Relazioni. Associazioni tra entità nel modello E-R.
- Derivazione del modello logico dallo schema E-R.
- Introduzione al DBMS.
- Normalizzazione delle relazioni.
- Integrità referenziale.
- Implementazione del modello fisico ed introduzione a Microsoft Access.

### Il linguaggio SQL:

- Caratteristiche del linguaggio non procedurale.
- Principali identificatori e tipi di dati.
- Definizione e manipolazione di tabelle.
- Interrogazione del database.
- Operazioni relazionali nel linguaggio: proiezione, selezione, congiunzione (join).
- Tipi di join (inner, outer, self join).
- Interrogazioni nidificate e complesse con tabella di appoggio.

- Funzioni di aggregazione.
- Ordinamenti e raggruppamenti.
- Condizioni di ricerca (like, between... and, in, any, all).

Tecniche di accesso ai database in ambiente Internet:

- Interfacciare un data base in rete.
- Programmazione lato server e lato client.
- Problematiche relative allo sviluppo di applicazioni web.

La sicurezza di un sistema informatico:

- La sicurezza dei dati in rete.
- Hacker e strumenti per violare la sicurezza.
- Crittografia.
- Cookie. Firewall.

La programmazione lato server:

- La piattaforma Apache-PHP-MySQL.
- Il linguaggio PHP.
- PHP e HTML.
- PHP e MySQL.

La produzione del software:

- Ciclo di vita del software e ingegneria del software.
- Analisi di problemi e documentazione.
- Classificazione del software.
- Software Open Source.

**Modalità di lavoro:**

1. Spiegazioni in classe e/o in laboratorio.
2. Utilizzo del laboratorio di Informatica.
3. Studio e svolgimento di esercizi a casa.
4. Assegnazione di compiti con stesura di relazioni sotto forma di siti web.
5. Recupero periodico.
6. Attività di approfondimento e ricerca.

**Strumenti di lavoro:**

1. Lavagna.
2. Proiettore di lucidi.
3. Testi in adozione.
4. Manuali di programmazione.
5. Laboratorio di Informatica.
6. Fotocopie di materiale didattico vario (articoli di riviste, appunti, schemi, ecc.).

**Tipologie di verifica:**

1. Verifiche formative orali, scritte, pratiche (se necessarie, su singole lezioni ed argomenti).
2. Verifiche sommative orali, scritte, pratiche.
3. Verifiche di recupero orali, scritte, pratiche (se necessarie).
4. Relazioni scritte in laboratorio o a casa.

**Numero di verifiche:**

Per ogni quadrimestre ogni studente sarà sottoposto mediamente a quattro prove di verifica sommativa di tipologia varia.

**Requisiti minimi per l'attribuzione della sufficienza:**

Lo studente:

- possiede i contenuti minimi: conoscenza delle problematiche di archiviazione e delle basi di dati, le basi del modello E-R, modello relazionale, elementi di base del linguaggio SQL, la sicurezza dei dati, basi della programmazione lato server, basi del linguaggio PHP
- applica le conoscenze acquisite per la soluzione di problemi noti (uso del dello E-R, derivazione dello schema relazionale, realizzazione di semplici applicazioni con basi di dati, anche lato server, uso del linguaggio SQL, uso del linguaggio lato server PHP, gestione elementare delle problematiche di sicurezza )

- si esprime in modo semplice ma con uso del linguaggio specifico della materia, cerca soluzioni anche precostituite.

Disciplina: **Informatica**

PIANO DI LAVORO Classe: 5<sup>^</sup> **I.T.I.S. corso Abacus**

Anno Scolastico 2009/10

PERIODO	CONTENUTI	OBIETTIVI	MODALITÀ	STRUMENTI	VERIFICHE	ORE
Settembre Ottobre Novembre	<ul style="list-style-type: none"> <li>- Approfondimenti del linguaggio Java.</li> <li>- Il sistema informativo aziendale</li> <li>- Introduzione alle basi di dati</li> <li>- Modello relazionale</li> <li>- Analisi di problemi</li> <li>- Il DBMS Microsoft Access: introduzione</li> <li>- Normalizzazione delle relazioni</li> <li>- Integrità referenziale</li> </ul>	<ul style="list-style-type: none"> <li>- Saper gestire le eccezioni nei programmi Java</li> <li>- Saper utilizzare i flussi in Java.</li> <li>- Conoscere i fondamenti delle basi di dati.</li> <li>- Conoscere il modello relazionale e le sue operazioni.</li> <li>- Conoscere sistemi per la gestione di basi di dati.</li> </ul>	1 - 2 - 3 - 4 - 5 - 6	1 - 2 - 3 - 4 - 5 - 6	1 - 2 - 3 - 4	65
Dicembre Gennaio Febbraio .	<ul style="list-style-type: none"> <li>- I linguaggi del DBMS</li> <li>- Il linguaggio SQL</li> <li>- Operatori relazionali</li> <li>- Interrogazioni del database</li> <li>- Aggregazioni, ordinamenti, raggruppamenti</li> </ul>	<ul style="list-style-type: none"> <li>- Conoscere ed utilizzare in diverse situazioni il linguaggio non procedurale SQL.</li> </ul>	1 - 2 - 3 - 4 - 5 - 6	1 - 2 - 3 - 4 - 5 - 6	1 - 2 - 3 - 4	65
Marzo Aprile Maggio Giugno	<ul style="list-style-type: none"> <li>- Il Database in rete. Il linguaggio PHP.</li> <li>- Sicurezza di un sistema informatico.</li> <li>- La documentazione del software</li> </ul>	<ul style="list-style-type: none"> <li>- Utilizzare Apache-MySQL-PHP.</li> <li>- Saper interfacciare un database in rete.</li> <li>- Conoscere le problematiche relative alla sicurezza ed adottare opportune cautele.</li> <li>- Saper analizzare i problemi in modo rigoroso ed efficiente</li> <li>- Conoscere il ciclo di vita del software</li> <li>- Conoscere i diversi tipi di software esistenti</li> </ul>	1 - 2 - 3 - 4 - 5 - 6	1 - 2 - 3 - 4 - 5 - 6	1 - 2 - 3 - 4	65

Disciplina: **Informatica – esercitazioni in laboratorio**

PIANO DI LAVORO  *Classe: 5<sup>^</sup> I.T.I.S. corso Abacus*

Anno Scolastico 2009/10

PERIODO	TEMA ESERCITAZIONI	OBIETTIVI	MODALITÀ	STRUMENTI	VERIFICHE	ORE
Settembre Ottobre Novembre	1. Creazione ed uso della Gui in Java. 2. Creazione ed uso delle applet 3. Uso dei flussi in Java  4. Esercitazioni in ambiente Microsoft Access con creazione tabelle, maschere, relazioni, query, report.	– Progettare e realizzare interfacce grafiche. – Progettare e realizzare applet anche complessi. – Progettare e realizzare applicativi che utilizzino i flussi  – Conoscere e saper utilizzare un sistema per la gestione di basi di dati.	1-2-3-4-5-6	1-2-3-4-5-6	1-2-3-4	33
Dicembre Gennaio Febbraio .	5. IL linguaggio DDL  6. Il linguaggio SQL	– Saper creare tabelle mediante il linguaggio DDL  – Conoscere ed utilizzare in diverse situazioni il linguaggio non procedurale SQL.	1-2-3-4-5-6	1-2-3-4-5-6	1-2-3-4	33
Marzo Aprile Maggio Giugno	7. Il linguaggio PHP  8. Realizzazione di applicativi completi con database in rete  9. Realizzazione sito web completo con programmazione lato client e lato server  10. Sviluppo di progetti per la sicurezza di un sistema informatico	– Saper realizzare applicativi in PHP – Saper utilizzare Apache-MySQL-PHP  – Saper interfacciare un database in rete.  – Approfondimenti e ripasso HTML, CSS, JavaScript  – Conoscere le problematiche relative alla sicurezza ed adottare opportune cautele.	1-2-3-4-5-6	1-2-3-4-5-6	1-2-3-4	33

## GRIGLIA DI DEFINIZIONE DEI REQUISITI MINIMI PER L'ATTRIBUZIONE DELLA SUFFICIENZA

<b>MATERIA:</b> informatica	<b>CLASSE:</b> 5 <sup>^</sup>	<b>INDIRIZZO:</b> informatica Abacus
-----------------------------	-------------------------------	--------------------------------------

<i>CONTENUTI</i>	<b>CONOSCENZE MINIME PER L'ACCESSO ALLA CLASSE SUCCESSIVA</b>	<b>COMPETENZE MINIME PER L'ACCESSO ALLA CLASSE SUCCESSIVA</b>
<ul style="list-style-type: none"> <li>• <i>Le basi di dati</i></li> </ul>	<ul style="list-style-type: none"> <li>• Conoscere le problematiche di archiviazione dati e delle basi di dati</li> <li>• Conoscere le basi del modello E-R e del modello relazionale</li> <li>• Saper usare del modello E-R per modellizzare semplici realtà d'interesse</li> <li>• Saper derivare lo schema relazionale</li> </ul>	<ul style="list-style-type: none"> <li>• Applica le conoscenze acquisite per la soluzione di problemi noti</li> <li>• Si esprime in modo semplice ma con uso del linguaggio specifico della materia, cerca soluzioni anche precostituite</li> </ul>
<ul style="list-style-type: none"> <li>• <i>Le basi del linguaggio SQL</i></li> </ul>	<ul style="list-style-type: none"> <li>• Conoscere le basi della programmazione con il linguaggio SQL</li> </ul>	
<ul style="list-style-type: none"> <li>• <i>Le basi del linguaggio PHP</i></li> </ul>	<ul style="list-style-type: none"> <li>• Conoscere le basi della programmazione lato server</li> <li>• Saper sviluppare semplici script con il linguaggio PHP</li> <li>• Saper applicare i concetti base della OOP a PHP</li> <li>• Saper realizzare semplici applicazioni con interfacciamento a una base di dati</li> </ul>	
<ul style="list-style-type: none"> <li>• <i>La sicurezza dei dati</i></li> </ul>	<ul style="list-style-type: none"> <li>• Conoscere i principali problemi legati alla sicurezza dei dati in rete ed ai possibili attacchi</li> <li>• Saper gestire in modo elementare le problematiche di sicurezza</li> <li>• Conoscere elementi di base della crittografia</li> </ul>	