IIS “G. Cigna G. Baruffi F. Garelli”	Programmazione individuale Piano didattico annuale

Materia	Informatica

Rif. Programmazione dipartimentale	*Segue la Progr. Del Dipartimento di “Matematica Applicata e Informatica ”*
Triennio	Amministrazione Finanza e Marketing, Articolazione Sistemi Informativi Aziendali
Classe	Classe 3^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.

<table>
<tr><td colspan="4">UNITA’ DI APPRENDIMENTO 1: Introduzione all’informatica : Storia delle Architetture e dei Sistemi operativi.</td></tr>
<tr><td>COMPETENZE:
Saper mappare l’architettura di un computer reale con un diagramma a blocchi (Arch. Von Neumann).</td><td colspan="2">OBIETTIVI SPECIFICI
✔ Comprendere passi fondamentali della lunga evoluzione Storica dei Calcolatori
✔ Comprendere le Componenti Hardware dei computer e il ruolo da esse svolto nell’architettura di Von Neumann
✔ Comprendere la necessità del Sistema operativo, le sue funzionalità, e la sua evoluzione</td><td>PERIODO:
12 Ore
Settembre ottobre</td></tr>
<tr><td>MACRO CONOSCENZE
Evoluzione nell’architettura delle macchine di ausilio al calcolo, di calcolo aritmetico e computer fino all’architettura di Von Neumann

Ricondurre il livello logico (cd “delle porte logiche) a un sottostante livello circuitale</td><td>CONTENUTO:
1. Evoluzione architetturale: da strumenti di ausilio al calcolo (Abaco), connessi all’adozione di un sistema di rappresentazione dei numeri posizionale additivo (meccanizzazione del calcolo, algoritmi delle operazioni), a macchine per il calcolo aritmetico (Pascal, Leibniz), fino all’idea di una macchina programmabile (Babbage e Ada Byron)
2. La Logica Matematica Proposta da G. Boole
3. Architettura di Von Neumann.
4. Evoluzione hardware: relè, valvola termoionica, transistor come elementi circuitali per implementare la logica e l’aritmetica binaria.
5. Possibili approfondimenti:
5.1. sviluppi futuri e trend recenti nelle ‘architetture di calcolatori
5.2. Cenni Storia Informatica (Contesto primi del ‘900)
6. Possibili sperimentazioni su sistema operativo (shell, linea di comando) eventualmente attraverso macchina virtuale</td><td>METODOLOGIA:
Lezione frontale, partecipata, con ausilio della proiezione dello schermo del PC alla classe si illustrano le tecniche applicative.

Lavori applicativi</td><td>TIPOLOGIA DI VERIFICA
Verifica orale, anche richiedendo l’esibizione di semplici abilità al calcolatore
Verifica con domande a risposta aperta
Test</td></tr>
</table>

UNITÀ DI APPRENDIMENTO **2 : - Elementi di LOGICA MATEMATICA**		
COMPETENZE: ◆ Applicazione della logica: Utilizzare le tecniche di logica per modellare situazioni reali nello sviluppo di software. ◆ Correttezza dei programmi.	**OBIETTIVI SPECIFICI** ■ Utilizzo della Logica Proposizionale e Di Boole , per modellare frasi del linguaggio naturale ■ Deduzione e derivazione di formule , a partire da talune e sulla base di certi fatti. ■ Utilizzare Formule logiche per l'implementazione di programmi, in particolare come strumento per implementare la logica del dominio desunta dai requisiti. ■ Formulare Espressioni logiche per comprendere l'esecuzione di algoritmi	**PERIODO:** Novembre Dicembre (10 ore)

MACRO CONOSCENZE	**METODOLOGIA:**	**TIPOLOGIA DI VERIFICA**

Teoria degli insiemi: Comprendere i concetti fondamentali come insiemi, sottogruppi e operazioni tra insiemi.
Calcolo proposizionale: Familiarizzare con le proposizioni, gli operatori logici e le equivalenze tra formule.
Logica dei predicati: Acquisire una conoscenza approfondita dei predicati e delle loro applicazioni.
CONTENUTO:
• Introduzione alla logica matematica , origini
• Ripasso di Prerequisiti Della teoria degli insiemi
• Logica Proposizionale
  • Proposizioni o Enunciati
  • Calcolo Proposizionale (operatori con le proposizioni tavole di verità degli operatori ∧ ∨ → ↔ ¬
  • Formule proposizionali, equivalenza tra formule.
  • Proprietà operazioni logiche
  • Regole di deduzione
• Logica dei Predicati
  • Predicati
  • Operazioni logiche con i predicati
  • Predicati e insiemi
  • Quantificatori (*esistenziale,universale)*
  • Implicazione Logica Equivalenza logica
	Lezione frontale partecipata (esercizi o interventi alla lavagna/lavagna digitale/tavoletta grafica)
	
	Testi e materiale digitali di supporto forniti dal docente per approfondire
		Verifica Scritta anche con domande aperte
		
		Test a risposta multipla
		
		Eventuale interrogazione
		
		Produzione di elaborati di sintesi degli argomenti affrontati

<table>
<tr><th colspan="4">UNITÀ DI APPRENDIMENTO 3 : INTRODUZIONE ALLA PROGRAMMAZIONE</th></tr>
<tr><td>COMPETENZE:
Realizzare Semplici Algoritmi con il paradigma imperativo strutturato, a partire da requisiti in linguaggio naturale. Con diagrammi di Flusso.
Realizzare Semplici programmi con il paradigma imperativo strutturato, a partire da requisiti in linguaggio naturale. Con linguaggio “Visuale” Flowgorithm</td><td colspan="2">OBIETTIVI SPECIFICI
Analizzare un Problema Algoritmico (riconoscerne i dati in ingresso, gli output da produrre e la relazione tra questi)
Comprendere e Produrre Diagrammi di Flusso .
Implementazione di semplici algoritmi in pseudo codice o linguaggi di programmazione (anche grafici) nel contesto della programmazione strutturata</td><td>PERIODO:

Ottobre Novembre (20 ore)</td></tr>
<tr><td>MACRO CONOSCENZE
Problema Algoritmico, analisi e specifica di algoritmi</td><td>CONTENUTO:
Problema Algoritmico e Algoritmo Risolvente
Possibile definizione di algoritmo
Formalismi in cui fornire la specifica di un algoritmo (pro e contro) Flow-Chart – Linguaggio Naturale - pseudo codice – Linguaggio di programmazione (nozione di “programma”)
Analisi del problema (dati in input, dati in uscita Output, relazione tra input e output)
Variabili e costanti (l’istruzione di assegnazione, tipo ,contenuto, identificatore di una variabile)
I Diagrammi di flusso (simboli grafici e significato)
Esercizi attraverso Diagrammi di flusso
Paradigma di programmazione Imperativo spaghetti coding e Paradigma di programmazione imperativo strutturato (Teorema Di Bohm Jacopini)
Gli Schemi di composizione fondamentali – Programmazione strutturata (sequenza-selezione-ripetizione)
Approfondimento sui cicli (condizionale, Post condizionale,, Ciclo For)</td><td>METODOLOGIA:
Lezione frontale partecipata (esercizi o interventi alla lavagna/lavagna digitale/tavoletta grafica)

Ulteriori Testi digitali di supporto forniti dal docente per approfondire</td><td>TIPOLOGIA DI VERIFICA
Verifica Scritta anche con domande aperte
Test a risposta multipla
Eventuale interrogazione
Produzione di elaborati di sintesi degli argomenti affrontati</td></tr>
</table>

<table>
<tr><th colspan="4">UNITÀ DI APPRENDIMENTO 4 : Le Variabili Strutturate e La loro Manipolazione</th></tr>
<tr><td>COMPETENZE:
◆ Tipi di dato strutturati o collezioni</td><td colspan="2">OBIETTIVI SPECIFICI
■ Individuare L'esigenza di utilizzare variabili strutturate
■ I tipi per rappresentare vettori e matrici</td><td>PERIODO:

Dicembre/Novembre
8 ORE</td></tr>
<tr><td>MACRO CONOSCENZE
● L’uso della memoria principale.</td><td>CONTENUTO:
• Variabili semplici e variabili strutturate
• Vettori e loro manipolazione
• L’elaborazione dei vettori
• Le matrici e loro manipolazione
• L’elaborazione di matrici
• Le stringhe di caratteri
• Esercitazioni in laboratorio Mirate (Flowgorithm programmazione imperativa e strutturata, vettori e stringhe)</td><td>METODOLOGIA:

• Esperienze in laboratorio, usando esercitazioni guidate

• Lezione frontale partecipata (esercizi o interventi alla lavagna/lavagna digitale/tavoletta grafica)</td><td>TIPOLOGIA DI VERIFICA
• verifica in laboratorio in cui si richiede all’allievo di raggiungere determinati obiettivi sul calcolatore
• verifica pratica al calcolatore
• test
• Verifica Scritta anche con domande aperte</td></tr>
</table>

<table>
<tr><td colspan="4">Unita’ di apprendimento5: PROBLEMI DI RICERCA E ORDINAMENTO E ALGORITMI FONDAMENTALI</td></tr>
<tr><td>COMPETENZE:
Individuare problemi notevoli e gestirne la soluzione</td><td colspan="2">OBIETTIVI SPECIFICI
■ Comprendere e conoscere i problemi di ordinamento e Ricerca, individuarli come sotto problemi di un problema dato
■ Conoscere le implementazioni degli algoritmi dei loro limiti per affrontare tali problemi</td><td>PERIODO:

Dicembre/Novembre
(8 ore)</td></tr>
<tr><td>MACRO CONOSCENZE

Complessità computazionale degli algoritmi</td><td>CONTENUTO:

• Problemi Notevoli di ricerca e ordinamento
• Algoritmi di ricerca (sequenziale, binaria)
• Cenni alla complessità computazionale
• Gli algoritmi di ordinamento
• Mirate sperimentazioni ((Flowgorithm programmazione imperativa e strutturata, implementazione di algoritmi di ricerca e ordinamento su vettori e stringhe)</td><td>METODOLOGIA:

• Esperienze in laboratorio, usando esercitazioni guidate

• Lezione frontale partecipata (esercizi o interventi alla lavagna/lavagna digitale/tavoletta grafica)</td><td>TIPOLOGIA DI VERIFICA
• Verifica pratica
• Verifica orale, anche richiedendo l’esibizione di semplici abilità al calcolatore
• Verifica con domande a risposta aperta
• Test</td></tr>
</table>

<table>
<tr><td colspan="4">UNITA’ DI APPRENDIMENTO 6: Sottoprogrammi E Ricorsione (paradigma procedurale)</td></tr>
<tr><td>COMPETENZE:
Scomporre un programma in più l sottoprogrammi</td><td colspan="2">OBIETTIVI SPECIFICI
• Strategie generali di modellazione alla soluzione dei problemi
• Comprendere la necessità dei sottoprogrammi
• Semantica della ricorsione</td><td>PERIODO:
Dicembre / Gennaio 5h</td></tr>
<tr><td>MACRO CONOSCENZE

Programmazione strutturata e Ricorsione</td><td>CONTENUTO:
• L’approccio top-down e bottom-up
• I sottoprogrammi
• Parametri formali e attuali
• Variabili Globali e Locali

Sperimentazione in laboratorio (Programmazione Procedurale in flowgorithm)
• Contenuti eventuali:
  • La ricorsione</td><td>METODOLOGIA:

• Esperienze in laboratorio, usando esercitazioni guidate

• Lezione frontale partecipata (esercizi o interventi alla lavagna/lavagna digitale/tavoletta grafica)</td><td>TIPOLOGIA DI VERIFICA
Verifica Strutturata
Ev Verifica Orale
Verifica Pratica
•</td></tr>
</table>

<table>
<tr><td colspan="4">UNITA’ DI APPRENDIMENTO 7: Introduzione a Linguaggi e strumenti per la Programmazione</td></tr>
<tr><td>COMPETENZE:

-Saper individuare e valutare caratteristiche di strumenti e linguaggi di programmazione per i propri scopi</td><td colspan="2">OBIETTIVI SPECIFICI

✔ Comprendere le fasi di sviluppo di un programma
✔ Compilatori Traduttori e Interpreti
✔ Caratteristiche dei linguaggi di programmazione</td><td>PERIODO:
Gennaio

(2Ore)</td></tr>
<tr><td>MACRO CONOSCENZE

• Processo di sviluppo software</td><td>CONTENUTO:

• I linguaggi di programmazione e i programmi traduttori
• Le fasi della programmazione (“l’IDE”)
• Classificazione dei linguaggi di programmazione (in base all’espressività offerta e ai paradigmi di programmazione supportati)</td><td>METODOLOGIA:

• Esperienze in laboratorio, usando esercitazioni guidate

• Lezione frontale partecipata (esercizi o interventi alla lavagna/lavagna digitale/tavoletta grafica)</td><td>TIPOLOGIA DI VERIFICA

Prove pratiche al calcolatore

Realizzazione di elaborati secondo specifiche e consegna.

Eventuale verifica orale.</td></tr>
</table>

<table>
<tr><th colspan="4">UNITÀ DI APPRENDIMENTO 8: Programmazione</th></tr>
<tr><td>COMPETENZE:
◆ Interazione con strumenti di Sviluppo integrati
◆ Sviluppo di Software</td><td colspan="2">OBIETTIVI SPECIFICI
■ Realizzazione di Programmi e documentazione in un moderno linguaggio di programmazione</td><td>PERIODO:
Febbraio- Maggio
(70 Ore)</td></tr>
<tr><td>MACRO CONOSCENZE
● Sviluppo di software
● Linguaggi di programmazione</td><td>CONTENUTO:
• 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:
5. Operatori aritmetici,di confronto, logici (connettivi in e not)..
6. Variabili, identificatori e tipi del linguaggio (linguaggio tipato dinamicamente → non necessaria dichiarazione di tipo)
7. L'output e l'input (le funzioni built-in input() print()).
8. il controllo del flusso condizionale (if – else , elif)
9. I cicli While (for visto quando si parla di stringhe e poi collezioni)
10. procedure e funzioni
• implementazione di Programmi semplificati di esigenze reali anche con spunto al dominio economico aziendale.
• Accenno ai Concetti di base del paradigma OO Oggetto,Classe, istanza
13. Le variabili strutturate e principali operazioni (scansione con for each)
14. Stringhe
15. liste, tuple (accennato)
Sperimentazioni: Su IDE Visual Studio, PyCharm</td><td>METODOLOGIA:
• Esperienze in laboratorio, usando esercitazioni guidate
• Lezione frontale partecipata (esercizi o interventi alla lavagna/lavagna digitale/tavoletta grafica)
• Libro di Testo U1,U2,U3,U4,U5,U6,U7,U8</td><td>TIPOLOGIA DI VERIFICA
• Test a risposta multipla
• Verifica scritta, anche con domande a risposta aperta
• Prevalentemente prova pratica al calcolatore
• Eventuale interrogazione</td></tr>
</table>