

Disciplina:

Anno scolastico: 20...../20.....

Classe/i :

Docente:

OBIETTIVI SPECIFICI DI APPRENDIMENTO

SEZIONE 1 Premesse matematiche

OBIETTIVI SPECIFICI

Nozioni fondamentali sui sistemi di numerazione
Sistemi di numerazione in base diversa da 10
Introduzione alla codifica delle informazioni
Concetti primari di logica matematica

Unità didattica 1 Sistemi di numerazione

Abilità

Rappresentare un numero con notazione polinomiale
Utilizzare la numerazione in base 2
Trasformare un numero da base 10 a base 2 e viceversa
Utilizzare la numerazione esadecimale
Trasformare un numero da base 10 a base 16 e viceversa
Trasformare un numero da base 2 a base 16 e viceversa
Eeguire le operazioni aritmetiche con basi diverse da 10.

Conoscenze

Notazione polinomiale dei numeri. Le cifre del sistema binario e del sistema esadecimale. Procedure di trasformazione da base 10 a base diversa da 10. Metodologia di calcolo aritmetico con basi diverse da 10.

Unità didattica 2 Codifica delle informazioni

Abilità

Applicare un codice univoco ad un insieme di oggetti
Codificare i caratteri che compongono le informazioni
Utilizzare il criterio di rappresentazione dei numeri interi all'interno di un elaboratore
Eeguire sottrazioni con il metodo del complemento a 2
Rappresentare i numeri reali in notazione esponenziale

Conoscenze

Caratteristiche del codice ASCII e UNICODE
Procedura per la rappresentazione binaria dei numeri interi
Definizione del complemento a due
Notazione in virgola mobile
Mantissa e caratteristica di un numero reale

Unità didattica 3 Algebra degli enunciati

Abilità

Usare le tabelle di verità
Rappresentare enunciati composti impiegando gli operatori logici
Riconoscere l'algebra degli enunciati come modello dell'algebra di Boole.

Conoscenze

Proposizioni ed enunciati
Gli operatori logici
Definizione dell'algebra di Boole

SEZIONE 2 Primi elementi di programmazione

OBIETTIVI SPECIFICI

Preparare l'ambiente di sviluppo
Acquisire familiarità con la struttura di un programma C#
Apprendere i primi elementi del linguaggio
Scrivere, compilare ed eseguire brevi programmi in C#

Unità didattica 4 Introduzione a C#

Abilità

Preparare l'ambiente di sviluppo
Definire una directory di lavoro
Utilizzare l'ambiente a riga di comando
Installare il .NET Framework
Utilizzare il comando di compilazione
Eseguire un programma C#
Compilare ed eseguire un semplice programma

Conoscenze

Come si accede all'ambiente a riga di comando
Come si esegue la compilazione di un programma C#
La struttura di un programma C#
Caratteristiche dell'ambiente .NET Framework
Definizione di codice sorgente e codice oggetto

Unità didattica 5 I tipi di dato in C#**Abilità**

Utilizzare il corretto tipo di dato
L'uso delle variabili
Dare il corretto nome alle variabili
Conoscere e scegliere tra i diversi tipi
Utilizzare i primi caratteri di escape
Utilizzare indipendentemente sia i tipi C# che quelli del .NET

Conoscenze

Concetto di tipo
L'uso dei tipi di variabili in relazione al problema
I tipi principali
I concetti di variabile e di costante
Differenza tra dichiarazione ed assegnazione
Caratteri di escape
I tipi di dati del Framework.NET

Unità didattica 6 Sistema di visualizzazione e di acquisizione**Abilità**

Gestire, in generale, una corretta visualizzazione dei risultati in ambiente console
Sapere utilizzare le istruzioni di acquisizione
Trattare il tipo string
Concatenare due o più stringhe
Convertire i dati

Conoscenze

Le istruzioni di stampa Write() e WriteLine()
I segnaposto nelle istruzioni di stampa
Le istruzioni di acquisizione Read() e ReadLine()
Come si effettua la concatenazione delle stringhe
L'uso dell'oggetto Convert

Unità didattica 7 Operatori**Abilità**

Scrivere espressioni aritmetiche
Comporre gli operatori con l'operatore di assegnazione
Verificare relazioni
Creare espressioni logiche

Conoscenze

I simboli per gli operatori aritmetici
Sintassi degli operatori aritmetici composti
I simboli per gli operatori relazionali
Tabelle di verità degli operatori logici

SEZIONE 3 Strutture di controllo**OBIETTIVI SPECIFICI**

Strutturare un algoritmo in modo logicamente corretto

Applicare i principi base della programmazione strutturata
Trasferire nel linguaggio di programmazione le strutture fondamentali di un algoritmo

Unità didattica 8 Algoritmi e pseudocodifica

Abilità

Analizzare un problema per la costruzione di un algoritmo
Realizzare i primi semplici algoritmi
Istruzioni di assegnamento, di ingresso e di uscita dei dati
Applicare i principi della programmazione strutturata
Scrivere un algoritmo utilizzando la pseudocodifica

Conoscenze

Definizione generale di algoritmo
Forma delle istruzioni di assegnamento, di input e di output in pseudocodifica
Teorema di Jacopini-Böhm
Pseudocodifica delle strutture di sequenza, alternativa e ripetizione

Unità didattica 9 - Istruzioni di controllo

Abilità

Usare le istruzioni di selezione
Utilizzare le diverse varianti dell'istruzione if
Organizzare la scelta multipla con switch
Realizzare semplici applicazioni con algoritmi strutturati
Lavorare con i connettivi logici

Conoscenze

Sintassi dell'istruzione if
L'operatore condizionale "?"
Come organizzare if nidificati
Sintassi dell'istruzione switch

Unità didattica 10 Istruzioni iterative

Abilità

Utilizzare le istruzioni di ripetizione nelle loro diverse forme
Sfruttare le istruzioni di controllo più complesse per la definizione di algoritmi complessi
Nidificare le strutture

Conoscenze

L'istruzione while
L'istruzione do.. while
L'istruzione for
Modalità di interruzione, di continuazione e di salto

SEZIONE 4 Metodi, array e strutture

OBIETTIVI SPECIFICI

Fornire ai programmi una struttura logica e organizzativa per facilitarne la leggibilità e la manutenzione
Suddividere un problema complesso in sottoproblemi più semplici
Saper isolare all'interno di un programma gli aspetti principali
Utilizzare schemi logici per organizzare insiemi complessi di dati
Saper classificare le modelli di dati in base al loro utilizzo
Presentare i risultati dell'elaborazione in modo rigoroso

Unità didattica 11 I metodi

Abilità

Organizzare in modo logico e funzionale un programma suddiviso in metodi
Utilizzare la sintassi per la dichiarazione di un metodo
Distinguere le variabili locali da quelle globali
Stabilire l'esatta collocazione delle variabili in ambiti funzionali
Utilizzare i valori di ritorno di un metodo
Impiegare metodi che usufruiscono di parametri
Gestire i parametri per valore, per riferimento e out

Conoscenze

Necessità dei metodi
Definizione e uso di metodi semplici
Ambito delle variabili: variabili locali e globali
Metodi: valori di ritorno
Sintassi per il passaggio di parametri
Passaggio di parametri per riferimento
Passaggio di parametri "out"

Unità didattica 12 Vettori, matrici e strutture**Abilità**

Definire un array a una dimensione in C#
Caricare un vettore in memoria
Definire in modo dinamico la dimensione di un vettore
Definire una matrice
Popolare una matrice
Visualizzare gli elementi di una matrice
Creare ed utilizzare le strutture
La memorizzazione avanzata dei dati in strutture
Utilizzare i metodi costruttori per le strutture di dati

Conoscenze

Concetto di vettore
Definizione di vettore
La struttura di controllo Foreach
Caricamento di un vettore in memoria
Dimensionamento dinamico
Le matrici
Dichiarazione di una struttura

Unità didattica 13 Manipolazione delle stringhe**Abilità**

Usare i metodi della classe String per la manipolazione delle stringhe
Formattare le informazioni e i dati di output
Utilizzare le stringhe di formattazione
Formattare le date e gli orari

Conoscenze

Le stringhe
Il metodo Concat
Il metodo Insert
I metodi ToLower e ToUpper
Le stringhe di formattazione
Il metodo ToString
Il metodo Format
Il metodo WriteLine(..)
Opzioni per formattare date e orari

SEZIONE 5 Classi e oggetti**OBIETTIVI SPECIFICI**

Risolvere un problema individuando gli oggetti e le loro interazioni
Definire una classe attraverso i suoi dati e i suoi metodi
Realizzare classi flessibili attraverso il polimorfismo
Strutturare gerarchie di classi sfruttando l'ereditarietà

Unità didattica 14 Concetti generali**Abilità**

Associare dati e codice in un'unica struttura
Definire più metodi individuati da una sola intestazione
Estendere dati e metodi da una classe di tipo generale ad altre classi derivate
Utilizzare un metodo standard per descrivere una classe

Conoscenze

Concetto generale di incapsulazione

Metodologia e sintassi per l'overloading
Concetto generale di polimorfismo
Definizione di derivazione e di ereditarietà
La terminologia e i diagrammi utilizzati nella OOP
Sintassi di base per la dichiarazione degli oggetti

Unità didattica 15 Polimorfismo ed ereditarietà

Abilità

Utilizzare i costruttori per definire un oggetto
Parametrizzare un costruttore
Individuare i membri static di una classe
Definire e gestire un array di oggetti
Utilizzare un unico identificatore per invocare metodi con funzioni simili
Definire classi polimorfe
Sfruttare la possibilità di derivare una classe da un'altra

Conoscenze

Sintassi dei costruttori parametrizzati
Definizione di un membro static
Metodologia per la definizione di array di oggetti
Sintassi per sfruttare l'overloading dei metodi e dei costruttori
Procedimento per ereditare in una nuova classe i membri di una classe più generale

SEZIONE 6 Visual C#

OBIETTIVI SPECIFICI

Sviluppare programmi visuali con l'uso di finestre e controlli
Realizzare applicazioni che prevedono l'interazione con l'utente
Mettere a disposizione dell'utente i controlli più appropriati in base alle necessità del programma
Guidare in modo efficace l'attività di chi utilizza l'applicazione
Prevedere e controllare le risposte dell'utente

Unità didattica 16 Visual Studio e C#

Abilità

Prendere confidenza con Visual C#
Utilizzare il form come unità base delle applicazioni a finestre
Inserire i controlli base
Associare ad un evento un gestore di evento
Scrivere il codice necessario per la gestione di un evento

Conoscenze

Le principali funzioni incorporate dall'IDE
Concetti basilari di una applicazione Windows
Come creare una applicazione basata su form
I controlli di base
Il concetto di evento
Gli eventi di uso più frequente

Unità didattica 17 Progetto di applicazioni user friendly

Abilità

Progettare applicazioni con interfaccia grafica
Inserire adeguati controlli in un form in relazione al progetto
Saper modificare le più importanti proprietà dei controlli
Aggiungere nuovi form al progetto

Conoscenze

Le caratteristiche dei principali controlli
Modalità di modifica delle proprietà di un controllo
L'impostazione delle proprietà principali dei controlli
Concetto di form modali e non modali

SEZIONE 7 Le Eccezioni

OBIETTIVI SPECIFICI

Ridurre al minimo le possibilità di blocco di un programma
Tenere sotto controllo eventuali errori dell'utente

Unità didattica 18 Controllo delle eccezioni

Abilità

Valutare gli errori che una applicazione può generare
Scrivere codice di gestione degli errori (eccezioni)
Utilizzare l'oggetto "eccezione"
Controllare la fine di un programma con anomalie
Costruire eccezioni personalizzate

Conoscenze

Sintassi per la gestione degli errori
sia in ambiente "console"
sia in un'applicazione windows
Modalità di utilizzo dell'oggetto "eccezione"
Metodo di la cattura di più eccezioni
Funzione del costrutto "try..catch..finally"

SEZIONE 8 Grafica

OBIETTIVI SPECIFICI

Utilizzare un form come superficie per il disegno
Disegnare forme geometriche
Inserire immagini

Unità didattica 19 La grafica in .NET Framework

Abilità

Definire una superficie per il disegno
Conoscere i principali metodi della classe Graphics
Inserire immagini in un oggetto grafico
Utilizzare l'oggetto Pen

Conoscenze

Modalità per il disegno di rettangoli
Modalità per disegnare ellissi
Metodi di gestione delle immagini
Caratteristiche dell'evento Paint

SEZIONE 9 I file

OBIETTIVI SPECIFICI

Apprendere il concetto di flusso
Capire la necessità degli archivi di dati
Rendere persistente lo stato di un oggetto: serializzazione

Unità didattica 20 I file

Abilità

Usare gli oggetti per la lettura di un file
Usare gli oggetti per la creazione un file
Registrare i dati su un file
Distinguere tra file di testo e binari

Conoscenze

I metodi che consentono di leggere informazioni da un file
I metodi che consentono di salvare informazioni in un file

Unità didattica 21 La serializzazione

Abilità

Comprendere il concetto di persistenza
Creare oggetti serializzabili
Ripristinare gli oggetti serializzati

Conoscenze

Il concetto di persistenza e serializzazione
Come rendere gli oggetti serializzabili
Come utilizzare le classi che permettono la persistenza