Programmazione c++

Obiettivi

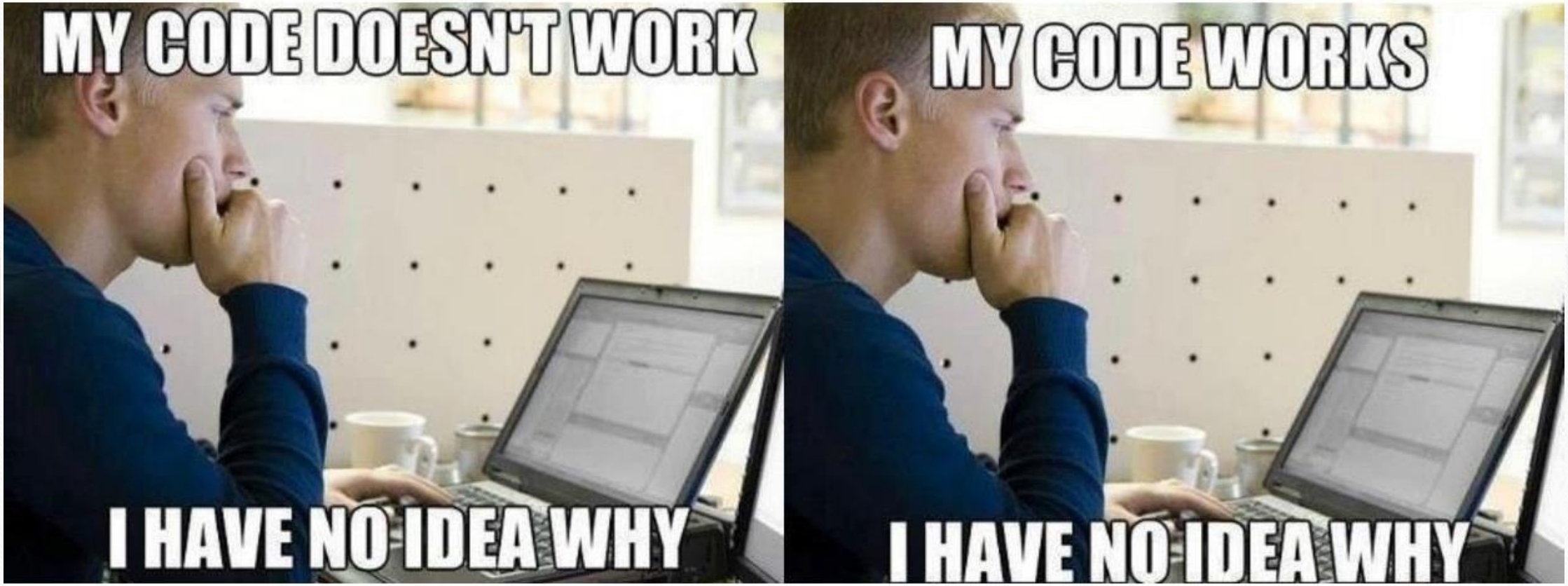
- Introdurre il linguaggio C++ e capire perché è stato scelto
- Conoscere i tipi di dati caratteristici del linguaggio
- Capire la logica degli operatori
- Codificare algoritmi in C++
- Usare le funzioni nella programmazione
- Usare gli array ed i vettori
- Usare le stringhe e le strutture di dati
- Capire i concetti principali delle classi
- Operare con gli archivi

La programmazione

Programmare oggi è una gara tra i tecnici del software che lottano per costruire programmi migliori ed a prova di idiota, e l'Universo che cerca di produrre migliori e più grandi idioti. Al momento, l'Universo sta vincendo.

La parte più difficile nella vita di un programmatore è quando si da la caccia ad un bug per una settimana, si trova il codice che genera il bug, si offende l'autore del codice ed infine ci si accorge di essere l'autore del codice maledetto

Se i costruttori costruissero come
i programmatori programmano,
il primo picchio che passa
potrebbe distruggere la civiltà


MY CODE DOESN'T WORK
I HAVE NO IDEA WHY
MY CODE WORKS
I HAVE NO IDEA WHY

Perché il c++?

Perché c++?

- Evoluzione naturale del C, da cui eredita tutta la sintassi
- Aggiunge il concetto di programmazione orientata agli oggetti
- E' uno dei linguaggi più diffusi nel mondo
- Essendo un linguaggio di «basso» livello, può essere utilizzato ovunque
- E' il più veloce per tempi di esecuzione

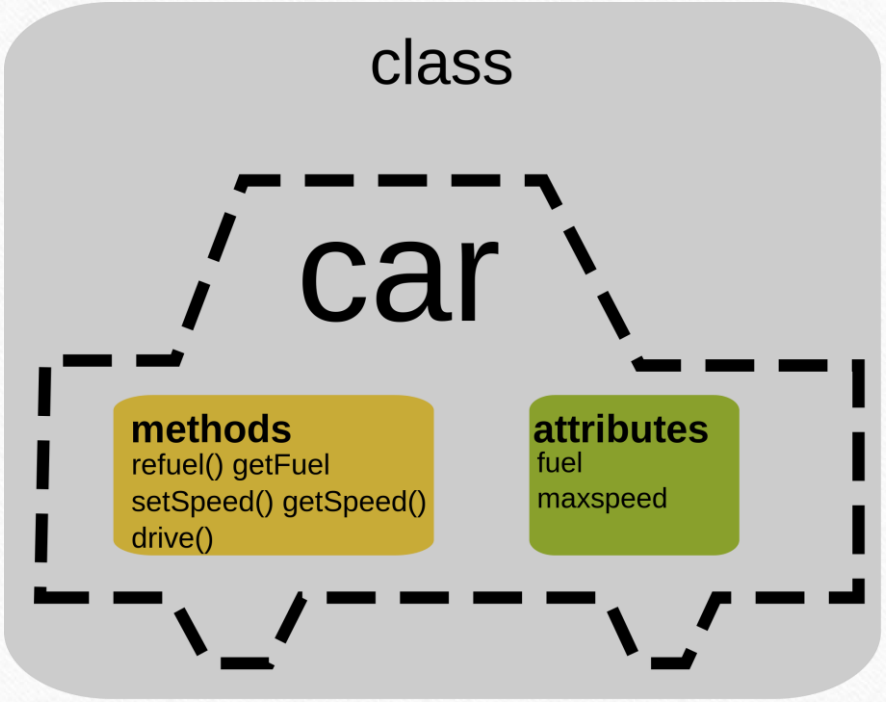
Usi del c++

Sviluppo di applicativi software

Sviluppo di driver per dispositivi informatici

Creazione di video games (es. Unity)

Programmazione ad oggetti

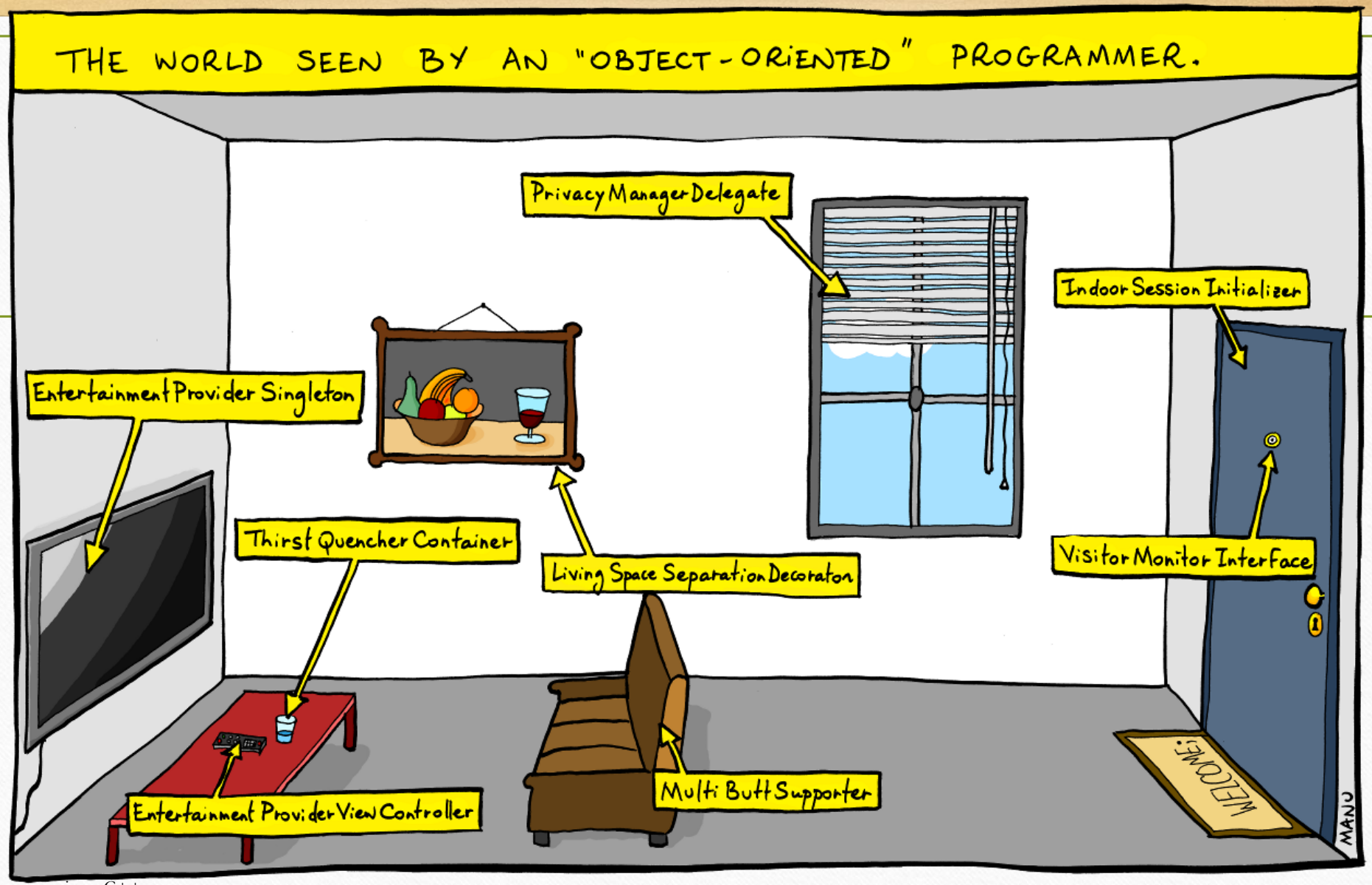


Programmazione ad oggetti

Pokemon
Name: Pikachu
Type: Electric
Health: 70
attack()
dodge()
evolve()

Fields

Methods


THE WORLD SEEN BY AN "OBJECT-ORIENTED" PROGRAMMER.
PrivacyManagerDelegate
IndoorSessionInitializer
EntertainmentProviderSingleton
ThirstQuencherContainer
LivingSpaceSeparationDecoraton
VisitorMonitorInterFace
EntertainmentProviderViewController
MultiButtSupporter
WELCOME
MANU

Miglioramenti principali rispetto al c

- Il miglioramento più importante, come già detto, è l'introduzione della programmazione ad oggetti.
- Altro miglioramento importantissimo sono i namespace, grazie ai quali è possibile separare variabili con nomi uguali
- È possibile dichiarare variabili contatore direttamente nei cicli for (Es. for(int i=0;i<..)).
- È possibile utilizzare i commenti di fine riga (//) oltre che i commenti multilinea (/* */).
- Introdotto il campo di visibilità con *const*
- Overloading delle funzioni (possibilità di utilizzare lo stesso nome per più funzioni)
- Pieno controllo della memoria da parte del programmatore con *new* e *delete*
- Introdotti i template, le funzioni generiche.

Riferimenti per il c++

- http://www.cplusplus.com/
- https://en.cppreference.com/w/
- https://cpppatterns.com/

Ambiente di programmazione

Dev-c++

Dev-C++ è un ambiente di sviluppo per chi programma in C/C++, che ti offre gli strumenti necessari per compilare, eseguire il debug e provare i programmi che hai sviluppato

La maggior parte dello schermo dedicato all'editor di testo. Uno spazio più limitato è dedicato al browser dei file e delle classi del progetto, e a monitorare i messaggi che arrivano dal compilatore e dal debugger

L'editor di testo offre parecchie opzioni interessanti in aggiunta alla classica colorazione sintattica, come l'auto-indentazione del codice, o la comparsa di tooltip che mostrano il prototipo di una funzione quando la si chiama

Dev-c++

Nel vostro desktop avete già l'icona del programma → apritelo

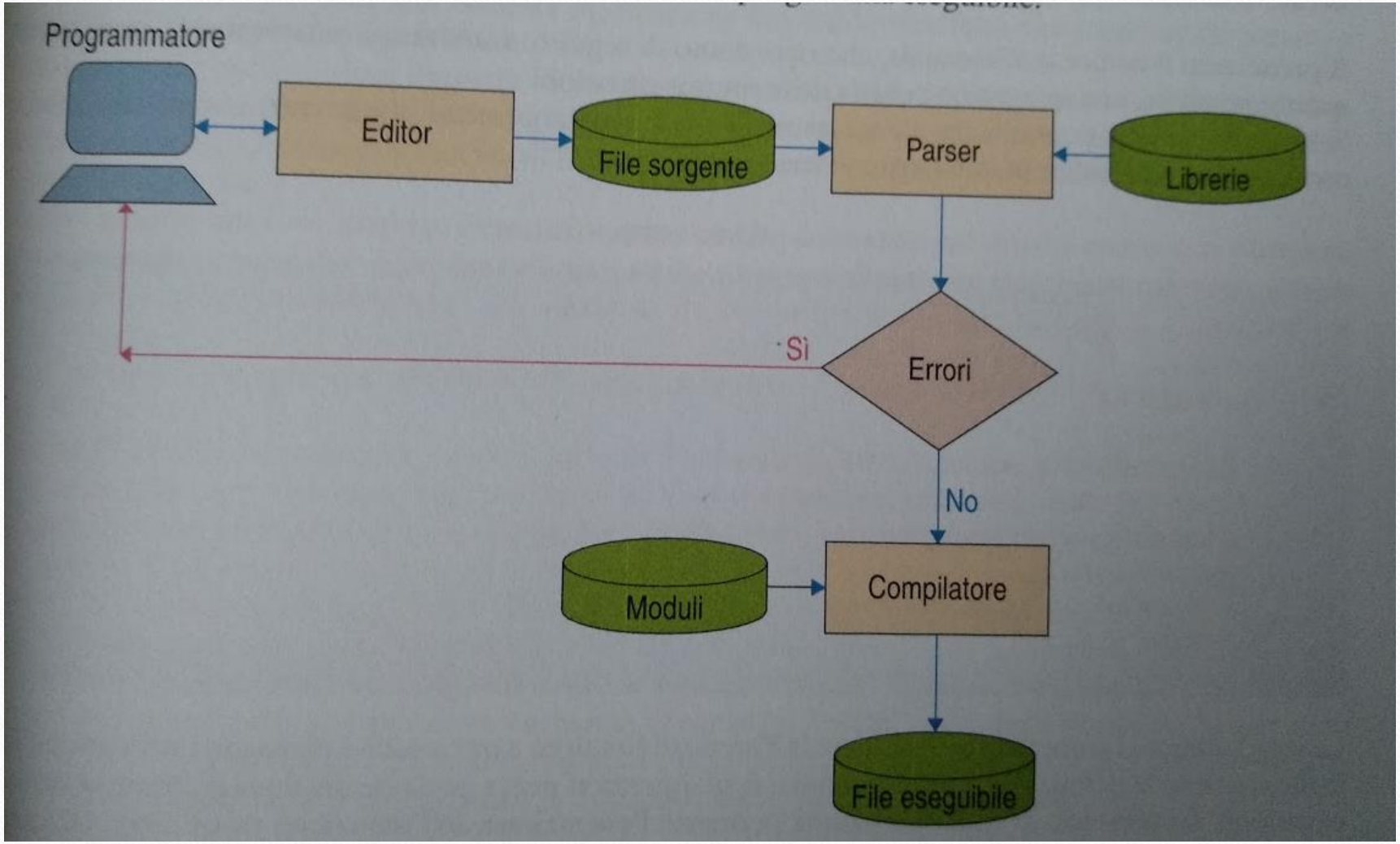
Contemplate l'interfaccia

Giocateci un attimo per capire i pulsanti principali

Date un'attenta occhiata a Strumenti → Opzioni

Dev-c++ contiene..

- L'editor di testi per creare il file sorgente
- Il **compilatore**, composto da
  - **Parser** = esaminatore di eventuali errori sintattici e grammaticali del linguaggio in cui è scritto il file sorgente
  - **Compilatore** vero e proprio, che trasforma il tutto in linguaggio macchina


Programmatore
Editor
File sorgente
Parser
Librerie
Errori
Si
No
Moduli
Compilatore
File eseguibile

Primo programma

```cpp
/*
    Primo programma che ci permette di rivivedere il dialogo tra Obi-Wan ed il Generale Grievous in Episodio III

    Questo programma ha lo scopo di introdurre gli elementi principali del linguaggio C++
*/

// includiamo una libreria che ci permette di leggere e stampare a video (cin e cout)
#include <iostream>

// in questo modo definiamo che stiamo usando lo standard C++, così che non dobbiamo sempre scirvere std::cout o std::cin
using namespace std;

//questa è la funzione principale, in pratica è da dove comincerà ad eseguire il codice
int main(){
    // cout ci permette di stampare a video
    cout << "Hello there!!" << endl;
    cout << endl;
    cout << "General Kenobi!!" << endl;
}
```

Ma che è tutta sta roba?

Analizziamo il codice assieme

Il codice potete trovarlo in Google Classroom, denominato come «Hello There.ccp»

Un file scritto in C++ deve sempre essere salvato con l'estensione .cpp

Per testarlo, scaricare il file, aprirlo con Dev-c++ e premere F11

Commenti

```
/*
    Primo programma che ci permette di rivedere il dialogo tra Obi-Wan ed il Generale Grievous in Episodio III

    Questo programma ha lo scopo di introdurre gli elementi principali del linguaggio C++
*/

// includiamo una libreria che ci permette di leggere e stampare a video (cin e cout)
```

Commenti

Sono utilizzati in genere come una descrizione delle funzionalità di un'istruzione o di un intero blocco di codice

Tutto ciò che serve per informazioni e/o descrizioni va inserito in un commento

Il compilatore li ignora, quindi servono solo a chi legge per essere informato sul programma

Commenti

Ne esistono di due tipi:

Commenti a riga singola, che iniziano con // e terminano alla fine della riga

Commenti delimitati, che iniziano con /* e terminano con */

#include

#include <iostream>

#include

Indica al compilatore di leggere la libreria di funzioni specificata e di inserirla nel codice

La libreria *iostream* include la definizione delle funzioni e degli operatori per la gestione dei flussi di immissione e di emissione → lettura e stampa da/a schermo

Nel nostro programma, in pratica, ci permette di usare ***cout*** e ***cin***

Funzione

```
int main(){
    //  cout ci permette di stampare a video
    cout << "Hello there!!" << endl;
    cout << endl;
    cout << "General Kenobi!!" << endl;
}
```

Funzione

Rappresenta un raccolta di istruzioni di programmazione che consente di portare a termine una determinata operazione

Ogni programma C++ deve disporre di una funzione main(), in quanto il compilatore va in cerca di questa funzione per iniziare il flusso di codifica

Funzione

Le istruzioni della funzione sono racchiuse tra le parentesi graffe

Le istruzioni vengono eseguite in sequenza, una riga alla volta

cout

```
cout << "Hello there!!" << endl;
```

```
cout << endl;
```

```
cout << "General Kenobi!!" << endl;
```

cout

Comando che scrive a video quanto indicato dopo

Se vogliamo stampare una stringa ben precisa, il testo da stampare deve essere incluso tra le virgolette

endl viene interpretato dal compilatore come un «vai a capo»

Formattazione testo con cout

CARATTERE	SIGNIFICATO
\a	Segnale sonoro
\r	Ritorno a capo
\v	Tabulazione verticale
\f	Avanzamento modulo (pagina)
\b	Backspace
\n	A capo (nuova riga)
\t	Tabulazione orizzontale
\\	Back slash
\’	Virgoletta singola
\”	Virgolette doppie

Carattere @

Se all'interno di una stringa dobbiamo visualizzare una \, ad esempio in

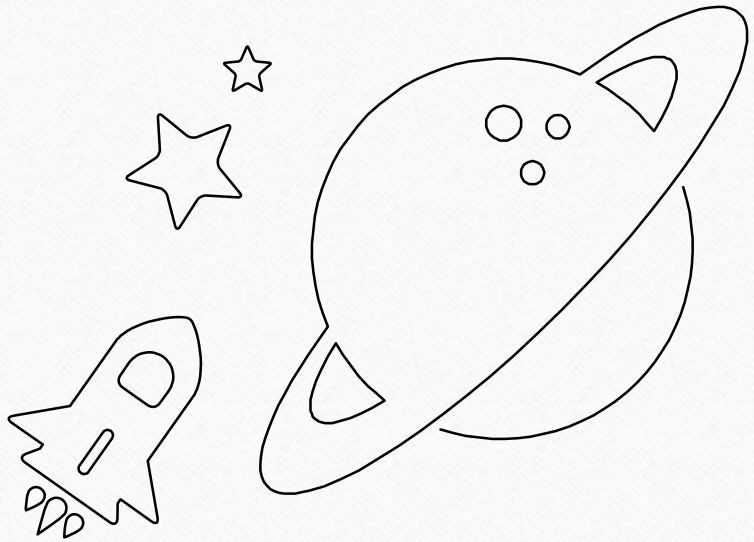
```
cout << = "c:\esempi\Error.txt";
```

Allora dobbiamo usare il carattere @ prima delle virgolette

```
cout << = @"c:\esempi\Error.txt";
```

Attenzione

Ogni istruzione C++ valida deve terminare con ; (punto e virgola), altrimenti si genera un errore di sintassi



Struttura di un programma C++

Esercizi

Modificare il codice fornito per fare in modo che stampi il vostro cognome invece di «Kenobi»

Scrivere un programma che stampi a video il messaggio «due volte» due volte con una riga di divisione

Esercizio – trova gli errori

```
/*
    Primo programma che ci permette di rivivedere il dialogo tra Obi-Wan ed il Generale Grievous in Episodio III

    Questo programma ha lo scopo di introdurre gli elementi principali del linguaggio C++
*/

// includiamo una libreria che ci permette di leggere e stampare a video (cin e cout)
#include <iostream>

// in questo modo definiamo che stiamo usando lo standard C++, così che non dobbiamo sempre scirvere std::cout o std::cin
using namespace std

//questa è la funzione principale, in pratica è da dove comincerà ad eseguire il codice
int main(){
    // cout ci permette di stampare a video
    cout << "Hello there!! << endl;
    cout << endl;
    cout << "General Kenobi!!" << endl;
}
```

cin

Acquisisce quanto digitato sulla tastiera fino a quando l'utente non preme il tasto invio

La sua esecuzione interrompe momentaneamente il flusso di elaborazione del programma e lascia il sistema in attesa dell'utente

RICORDA!! Il ***cin*** deve sempre salvare quanto riceve in una variabile

cin

cin >> numero

Aspetta che l'utente inserisca un valore e lo salva nella variabile numero

Regola di buon uso

Sempre mettere un *cout prima di un* cin, specificando all'utente quello che si vuole che venga scritto

Esempio

Scarica il file «Conosciamoci.cpp» da Google Classroom

Aprilo con Dev-C++

Analizzalo per bene

Eseguilo premendo F11

Esercizio

Aggiungere al programma precedente l'inserimento del cognome, con relativa stampa

Aggiungere al programma precedente l'inserimento dell'età, con relativa stampa