

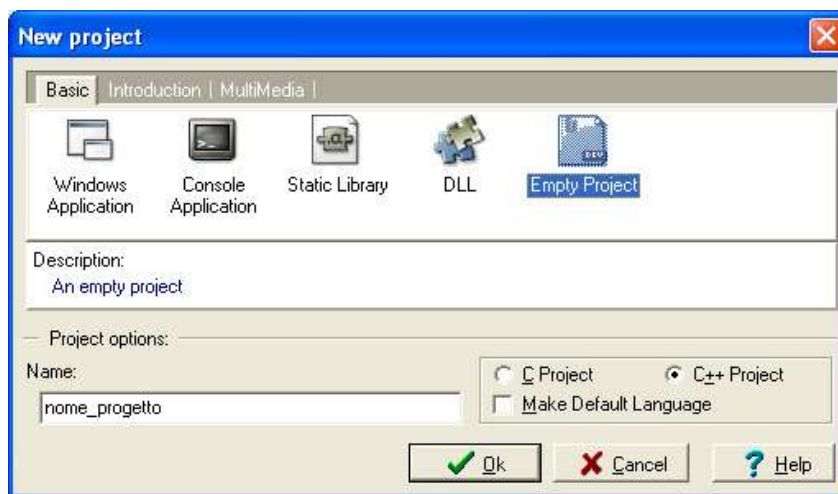
Guida all'uso dell'ambiente di sviluppo¹ integrato o IDE Dev-C++ versione 4.9.9.2

Inizializzazione: creazione del nuovo *progetto* e del *file sorgente* in ambiente Dev - C++

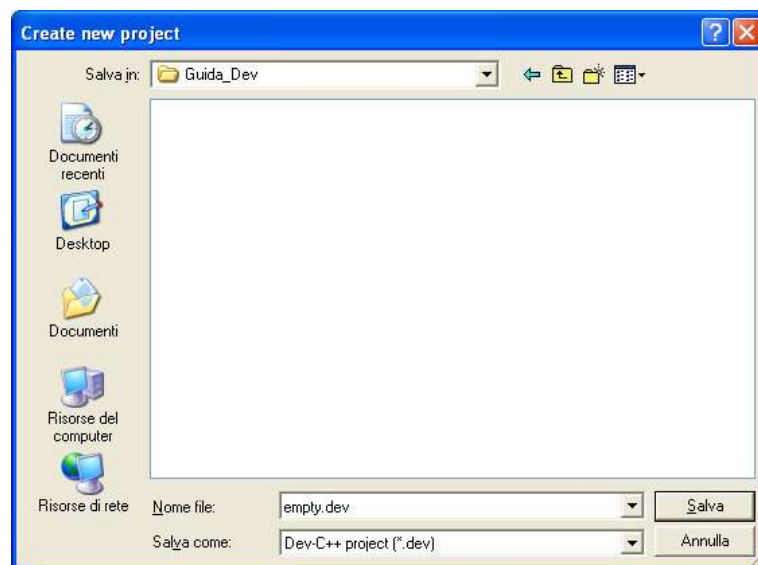
Al lancio del programma si apre la finestra tipica delle applicazioni in ambiente Windows, che presenta la barra dei menù:

File Edit Search View Project Execute Debug Tools CVS Window Help

- Selezionando una prima volta i comandi **File** → **New** si scelga **Project**
e si crei il progetto di tipo **Win32 Console Application** scegliendo un nome significativo

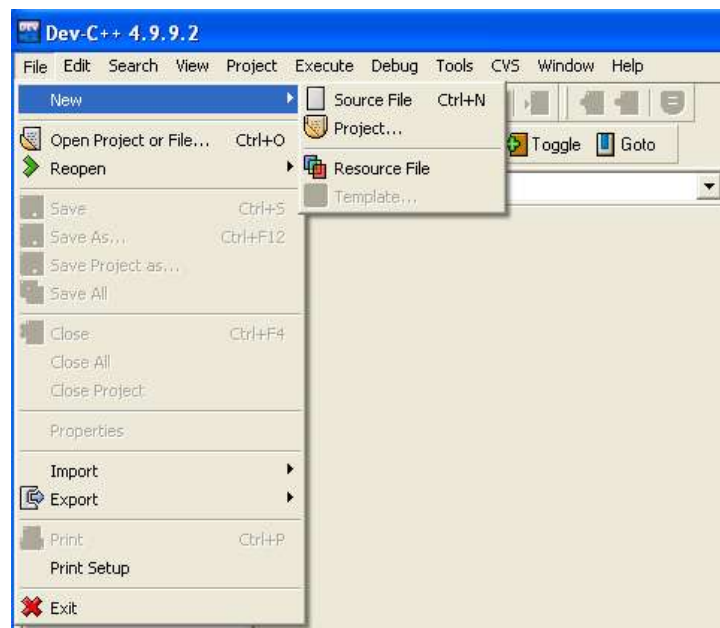


La creazione guidata del progetto *empty* richiede il salvataggio con scelta del corretto percorso.



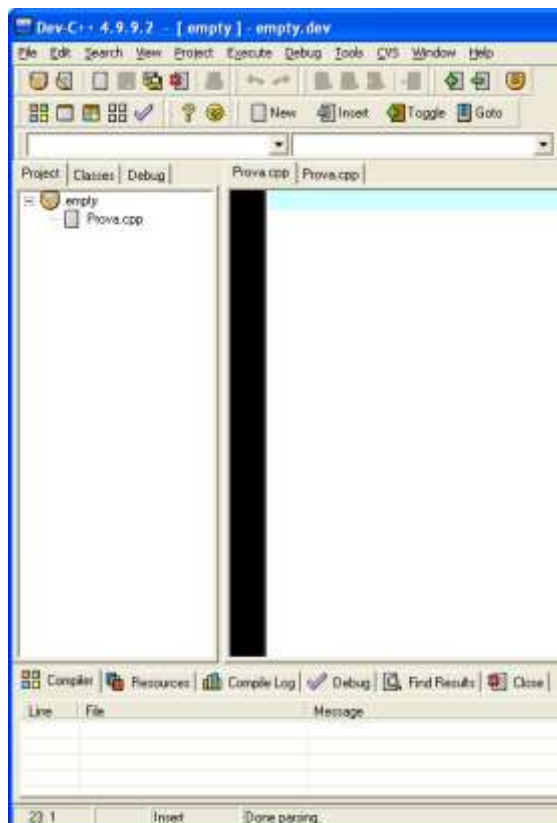
¹ Tra i livelli di funzionalità di un calcolatore si definisce **software di sviluppo** quell'insieme di programmi che consentono di sviluppare le applicazioni definite dall'utente: editor, compilatori, interpreti e linker-loader cioè programmi capaci di tradurre sequenze di istruzioni formulate in linguaggio di programmazione in sequenze di numeri in un codice comprensibile dal calcolatore. Tale software è un esempio di software di base

- Selezionando una seconda volta i comandi **File** → **New** si scelga **Source File** che crea automaticamente un file di tipo C++ da salvare con un nome significativo



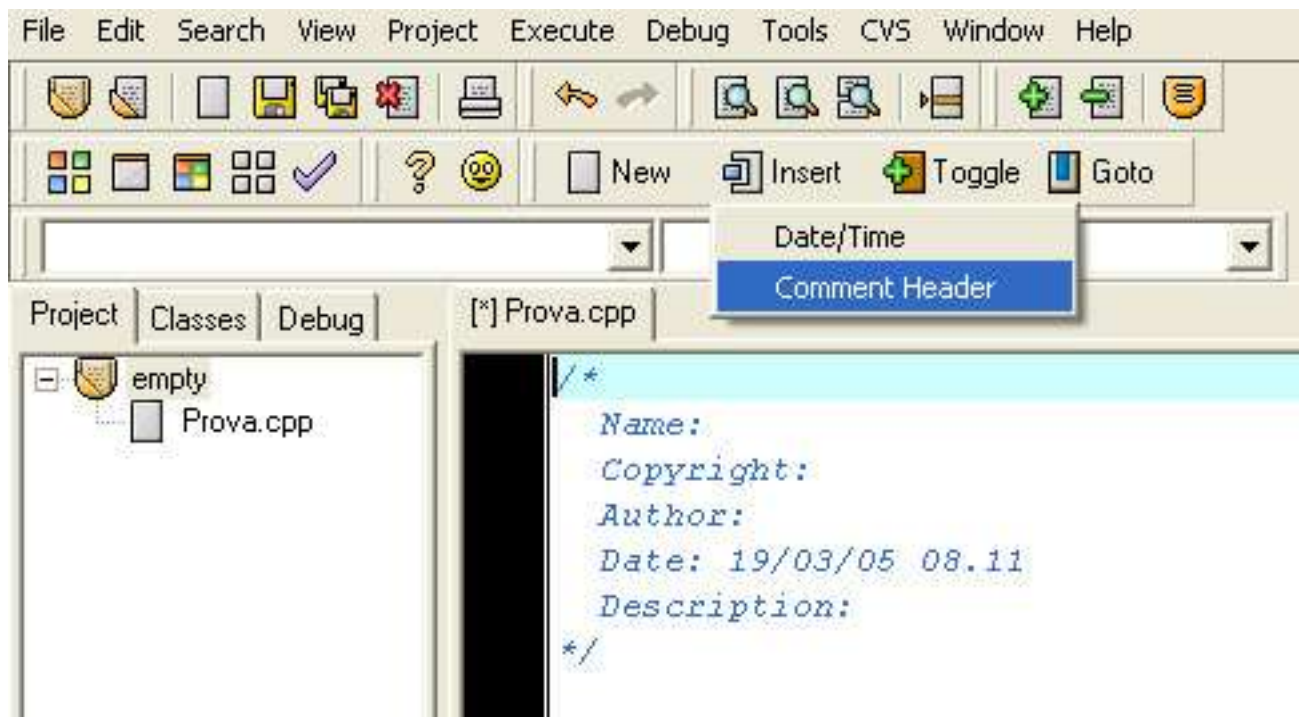
Prima FASE: uso dell'**editor integrato** per scrivere il *file sorgente*

Per “default” sono visualizzati sia l’ambiente di progetto, sia l’Output per informazioni sull’esito delle varie fasi nella costruzione del programma, sia il foglio in cui scrivere.



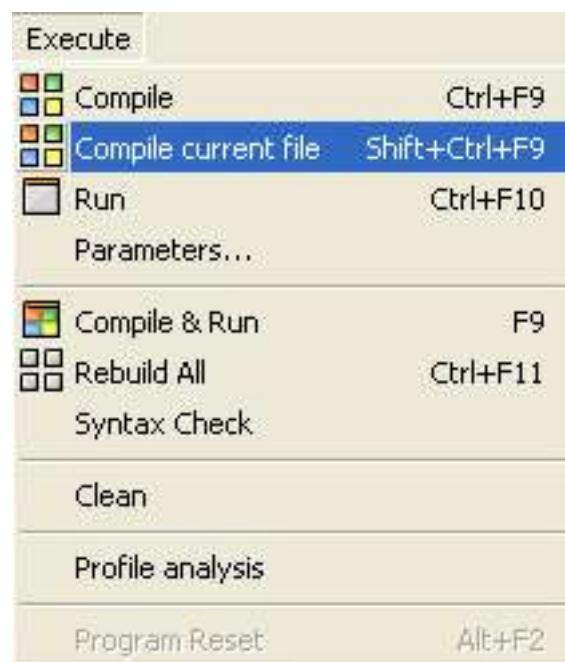


NB: E' **buono stile** di programmazione, inserire gli opportuni *commenti* a partire da una tipica intestazione come quella visualizzata selezionando l'opportuna opzione di inserimento



Seconda FASE: compilazione (traduzione in esadecimale con controllo della sintassi del linguaggio C/C++)

- Selezionando il comando **Execute** → **Compile current file**



NB: la funzione **main** deve ritornare un valore **intero** poiché l'**ambiente inserisce automaticamente** l'istruzione **return 0;**

Terza FASE: linking cioè collegamento con librerie e indirizzamento (*mappatura indirizzi*) per caricare (**loader**) in Ram

- Selezionando il comando **Execute** → **Compile**



Si possono, in alternativa, usare le opportune **icone**:

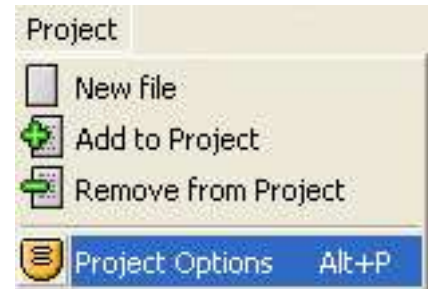



- Per realizzare la fase di **COMPILAZIONE** e **LINKING**
- Per **ESEGUIRE** velocemente
- Per realizzare le fasi di **COMPILAZIONE**, di **LINKING** ed **ESEGUIRE** velocemente
- Per ricostruire (**rebuild all**)
- Per **DEBUG**: esecuzione passo-passo

ATTENZIONE: unica funzione principale *main*

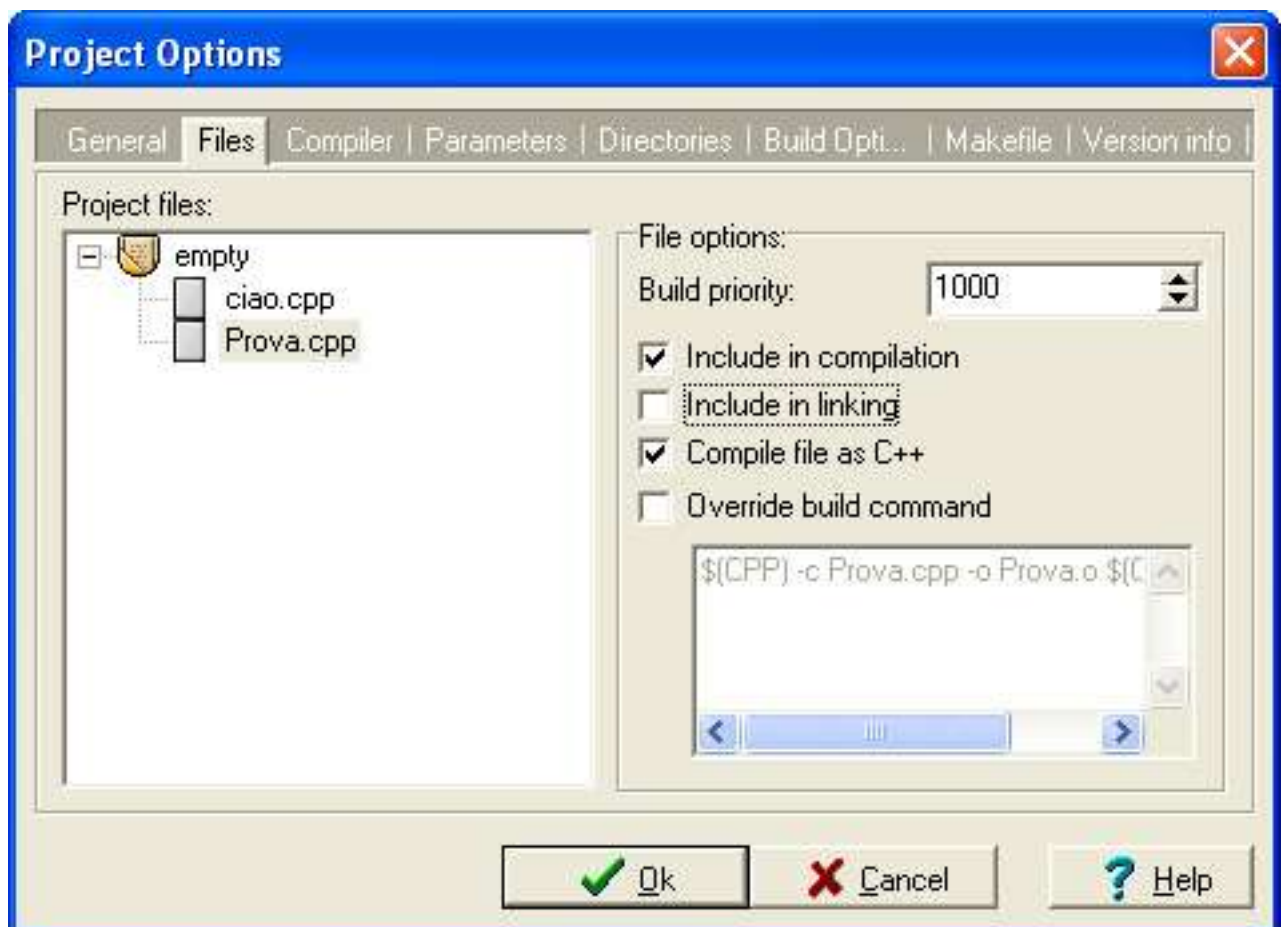
In ogni progetto si possono inserire più programmi (contenenti la funzione principale **main**) ma solo uno può essere selezionato in fase di costruzione. Per deselegionare² gli altri ed escluderli dalla fase di linker-loader:


- Selezionare da menu il comando **Project → Project Options → Files**



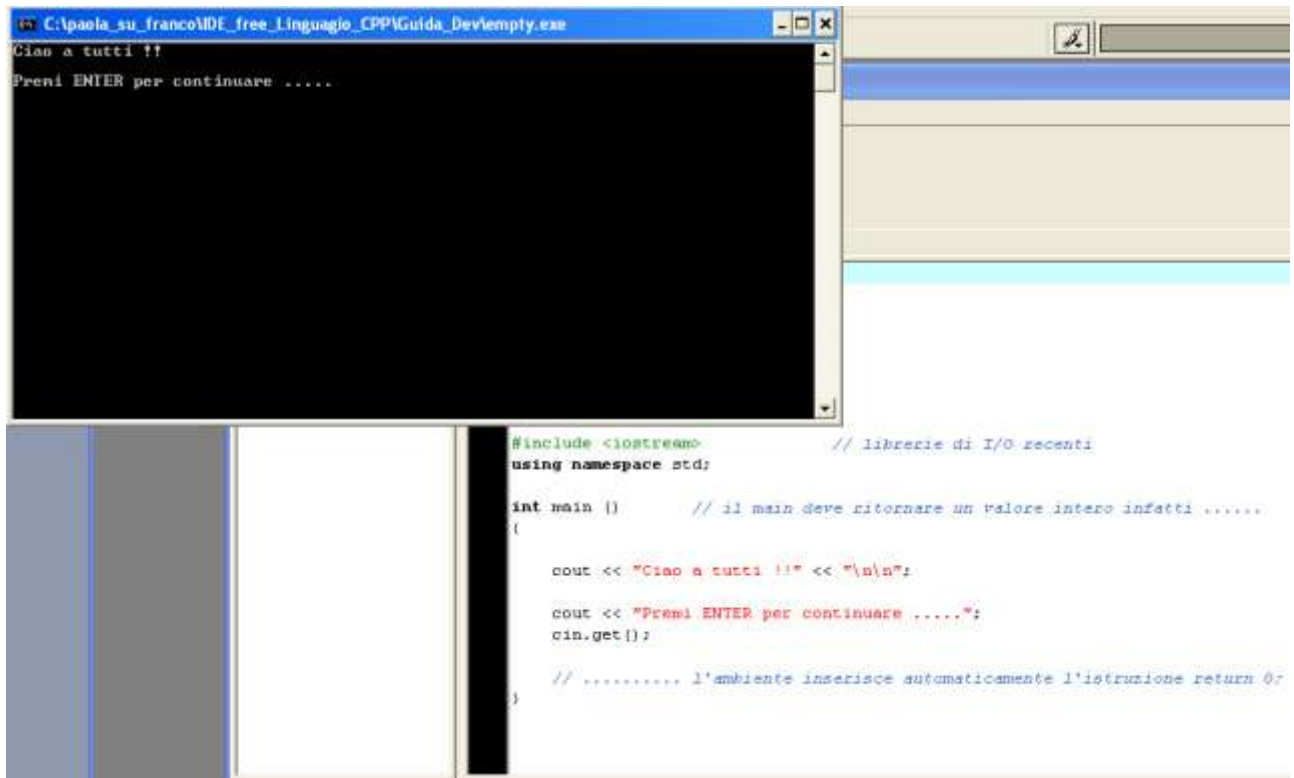
oppure usare l'icona opportuna  selezionando **Files**

- Selezionare il file e smarcare la casella per escluderlo dalla fase di linker-loader:



² Alternativa possibile è **rimuoverli**  temporaneamente dal progetto escludendoli anche dalla fase di compilazione (operazione possibile anche selezionando il file e con tasto destro attivando da menu pop_up il comando di rimozione)

ESECUZIONE veloce: si apre una finestra console se si inviano messaggi al monitor



NB: E' necessario:

- adoperare la sintassi più evoluta per le direttive di precompilazione³, usando come "contenitore" il **namespace std**

```
#include <iostream> // nome della libreria di I/O e non nome del file  
using namespace std;
```

- provvedere a mantenere aperta la finestra console (ad esempio fino alla pressione del tasto INVIO)

```
cout << "Premi ENTER per continuare ....."  
cin.get();
```

Collaudo: Il modo più facile di verificare la funzionalità di un algoritmo è quello di tradurlo in programma per un sistema di esecuzione reale e provarlo su calcolatore. In ambiente Dev-C++ esistono facilitazioni per effettuare tale test come la possibilità di avere una traccia passo-passo delle elaborazioni svolte e dei valori assunti dalle variabili.

>>

³ Direttive che non codificano un comportamento del programma ma si limitano a dare istruzioni di servizio al compilatore per la generazione del file eseguibile