

I TUTOR

Massimiliano Cattafi

e-mail:

massimiliano.cattafi@student.unife.it

Ilaria Laurenti

e-mail:

ilaria.laurenti@student.unife.it

IL LABORATORIO

- Perché il laboratorio?
 - Alcune problematiche si capiscono meglio nell'uso pratico
 - E' importante acquisire manualità nell'uso dell'elaboratore



Il laboratorio fa parte integrante delle lezioni

- ... e chi ha un PC a casa?
 - Può svolgere le esercitazioni a casa (ma in lab c'è anche il tutor ...)

IL LABORATORIO

- Esercitazioni guidate:
 - **Mercoledì mattina e pomeriggio**
 - Di volta in volta verranno presentate le soluzioni dell'esercitazione precedente ed i nuovi esercizi.
- Esercitazioni non guidate:
potete accedere al laboratorio quando volete se non è occupato da altre lezioni (e c'è posto)
- Lucidi delle esercitazioni:
 - disponibili sul sito del corso:
`http://www.unife.it/ing/informazione/fondamenti-info-1/materiale-didattico/esercitazioni-in-laboratorio/`

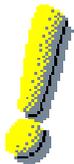
IL LABORATORIO

- Labinfo all'ultimo piano
- 64 PC con Windows 2000 Professional
- Per entrare bisogna inserire le informazioni del proprio account:

username: `utente`

password: `Infonew1`

- Sull'Hard Disk, si può scrivere solo sulla directory C:\TEMP, che viene cancellata ogni volta che si entra.



Per memorizzare i programmi/dati è necessario portare un floppy disk o una chiavetta USB

AMBIENTI DI SVILUPPO

In laboratorio utilizzeremo l'ambiente integrato Visual C++ da Visual Studio 2005, per esercitarvi a casa potete richiedere una copia di MS Visual Studio 2005 in licenza accademica (gratis).

Se avete scritto il vostro indirizzo e-mail nel questionario distribuito in aula, vi arriverà un'e-mail con le istruzioni per il download (in caso di problemi di qualche tipo fateci sapere).

Sul sito del corso (materiale didattico->software) sono consigliati anche altri ambienti di sviluppo.

Ad esempio Dev-C++ è free software

<http://www.bloodshed.net/devcpp.html>

Si può anche usare il compilatore da riga di comando gcc (come vedremo nelle prossime slide)

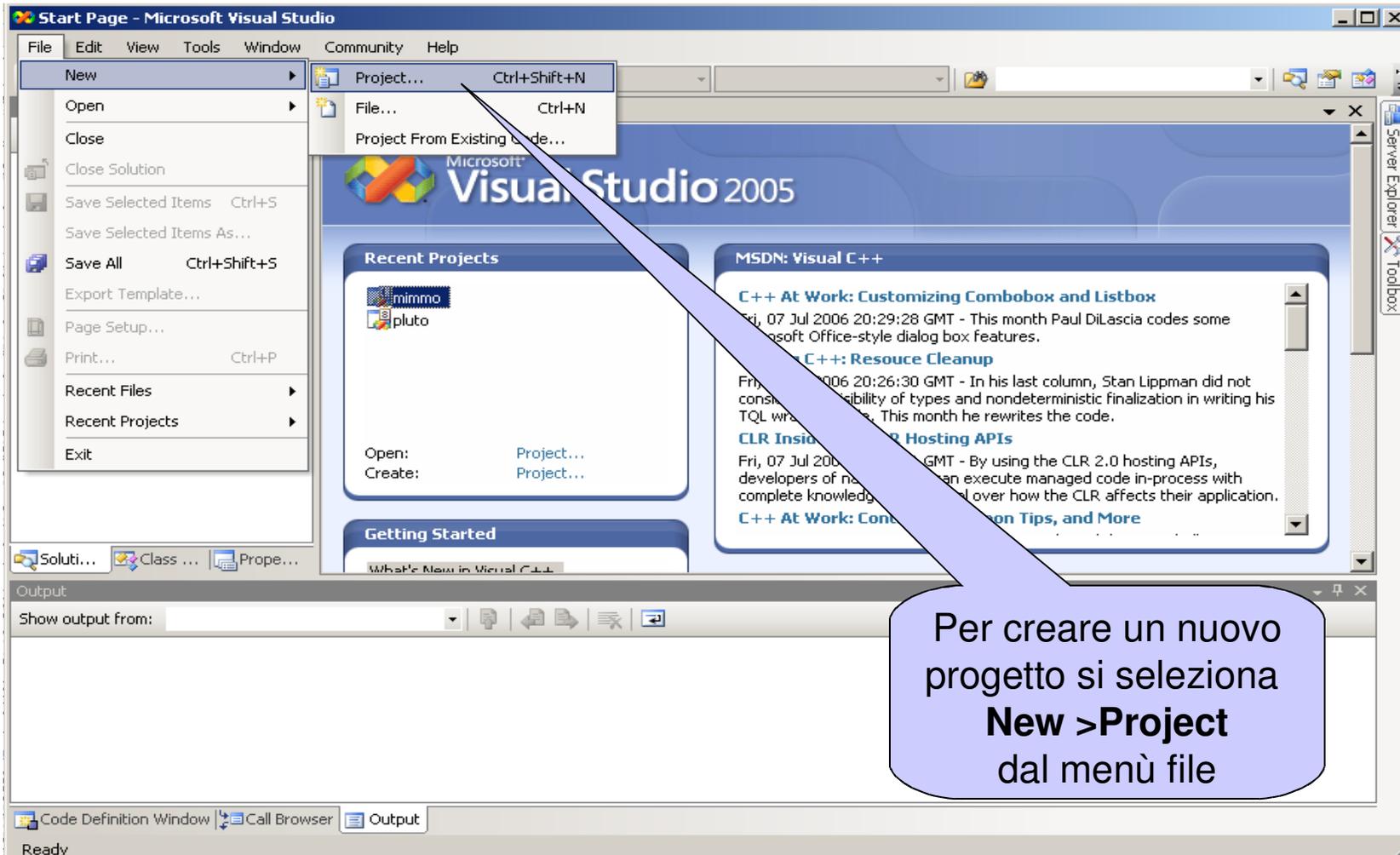
MS Visual C++

Fa parte del pacchetto Microsoft Visual studio, che contiene vari strumenti per lo sviluppo di software: Visual J (Java), Visual Basic, VSS (Visual Source Safe, tool di versionamento)...

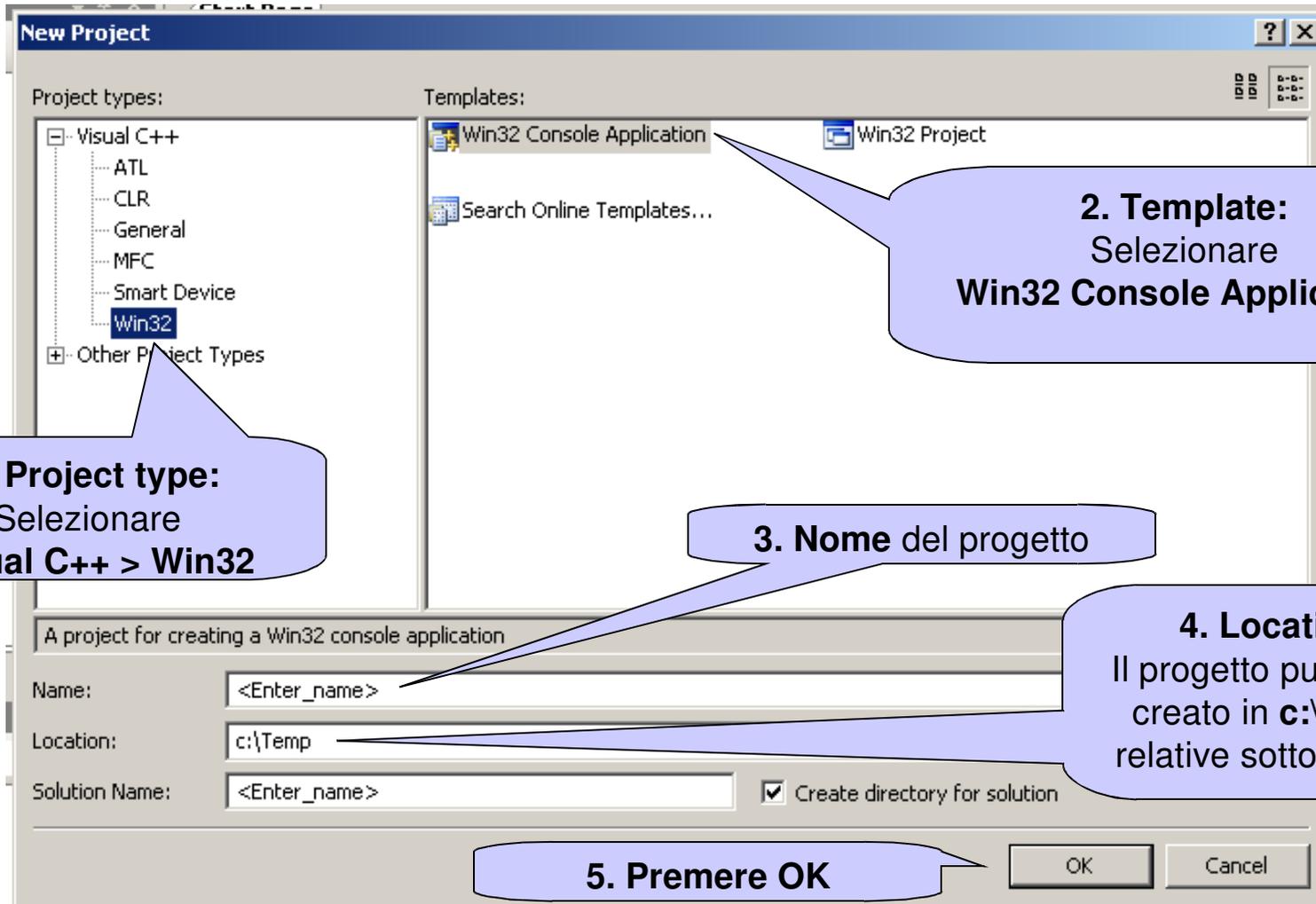
Viene utilizzato principalmente per sviluppare in C++, in questo corso verrà utilizzato soltanto con il C.

Vediamo ora come creare un semplice progetto.

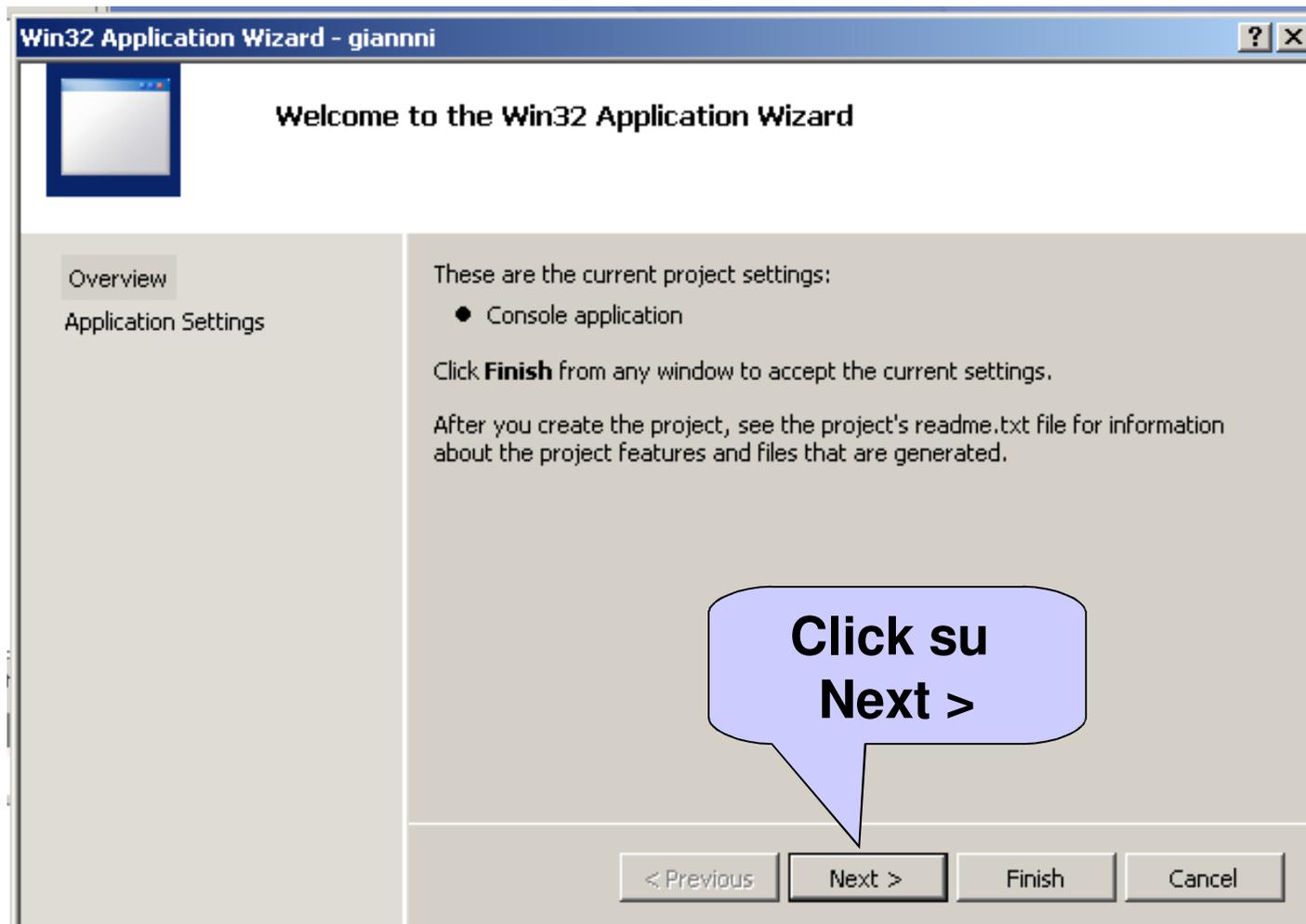
Creare un nuovo progetto



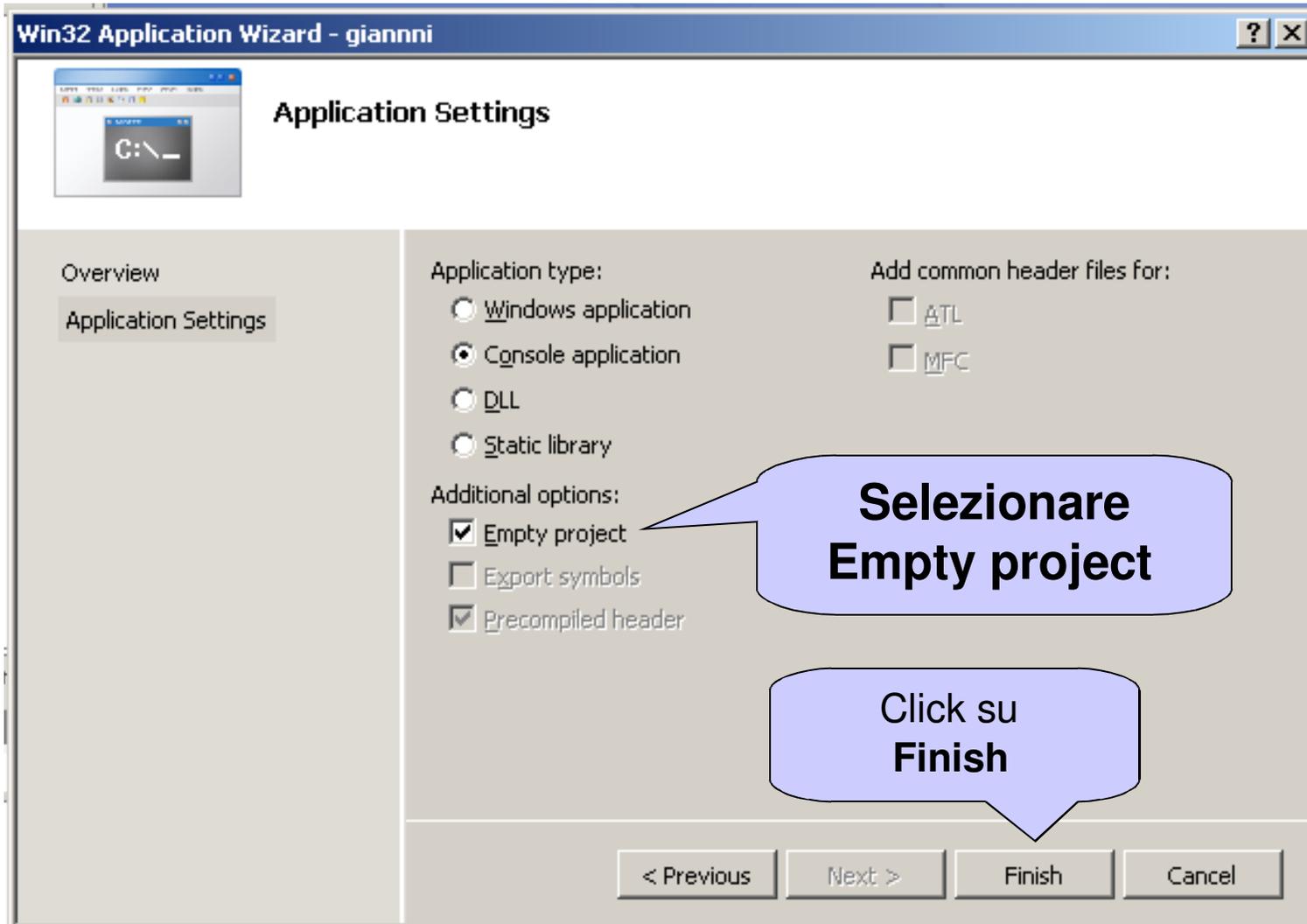
Creare un nuovo progetto



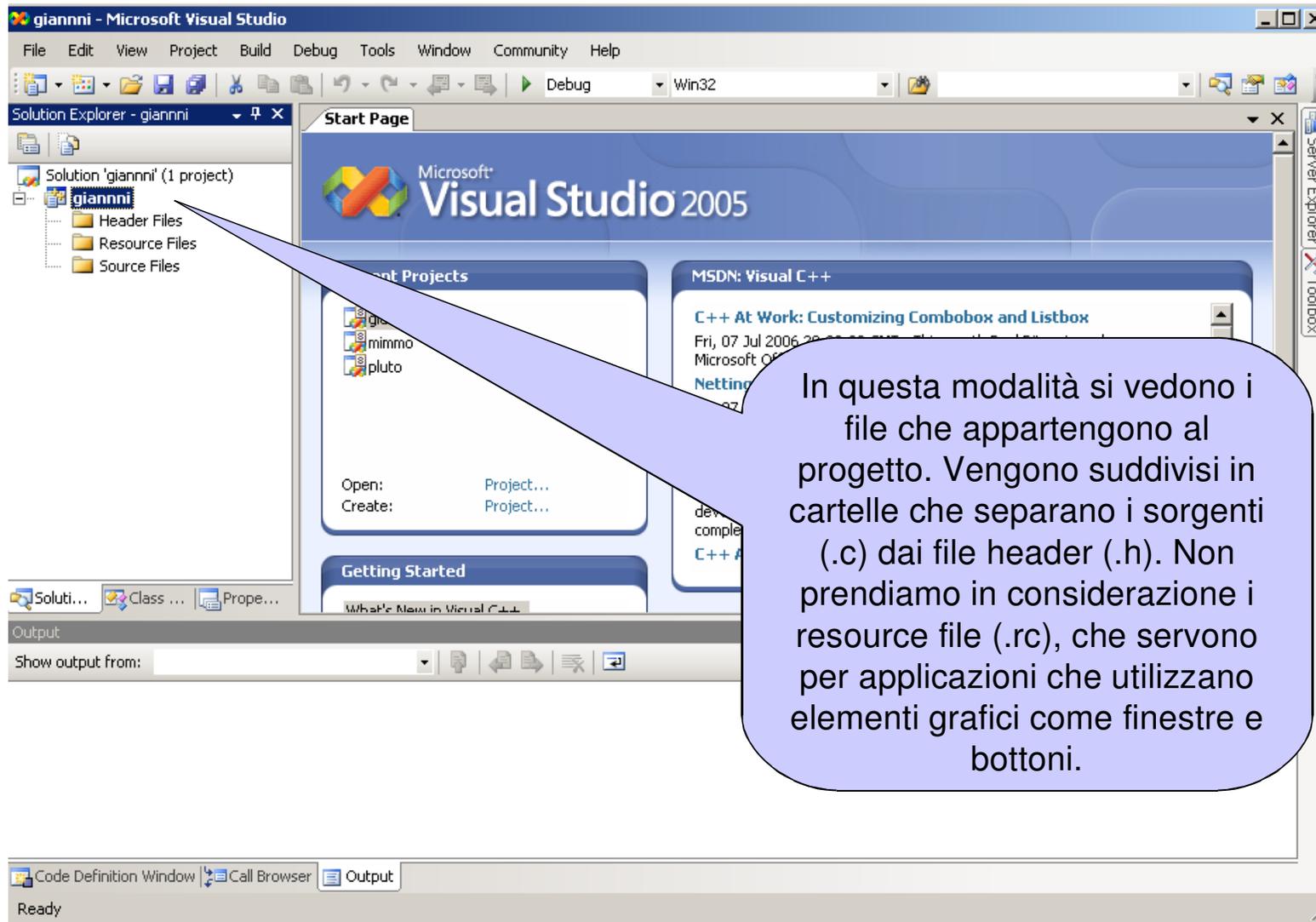
Creare un nuovo progetto



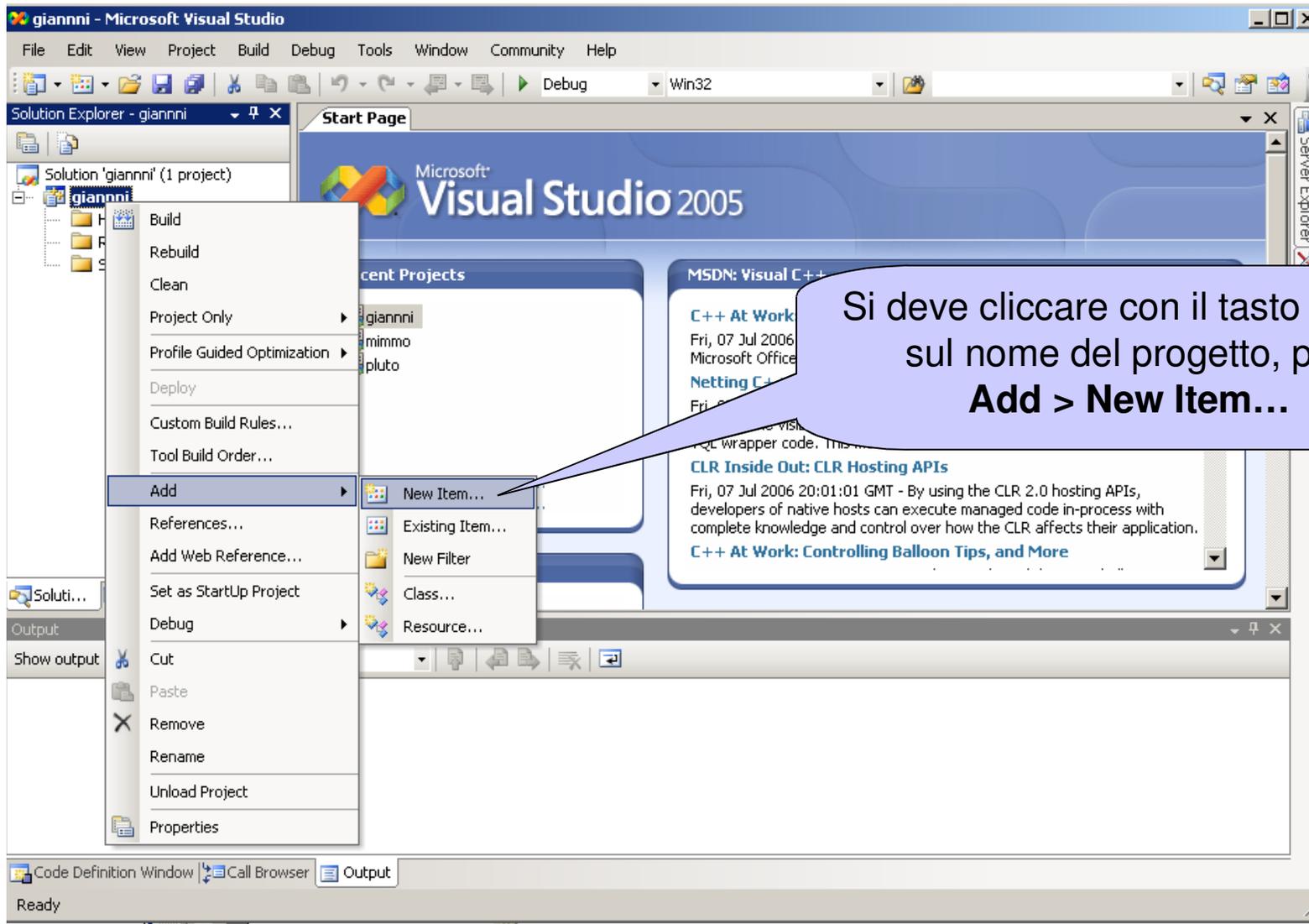
Creare un nuovo progetto



Solution Explorer

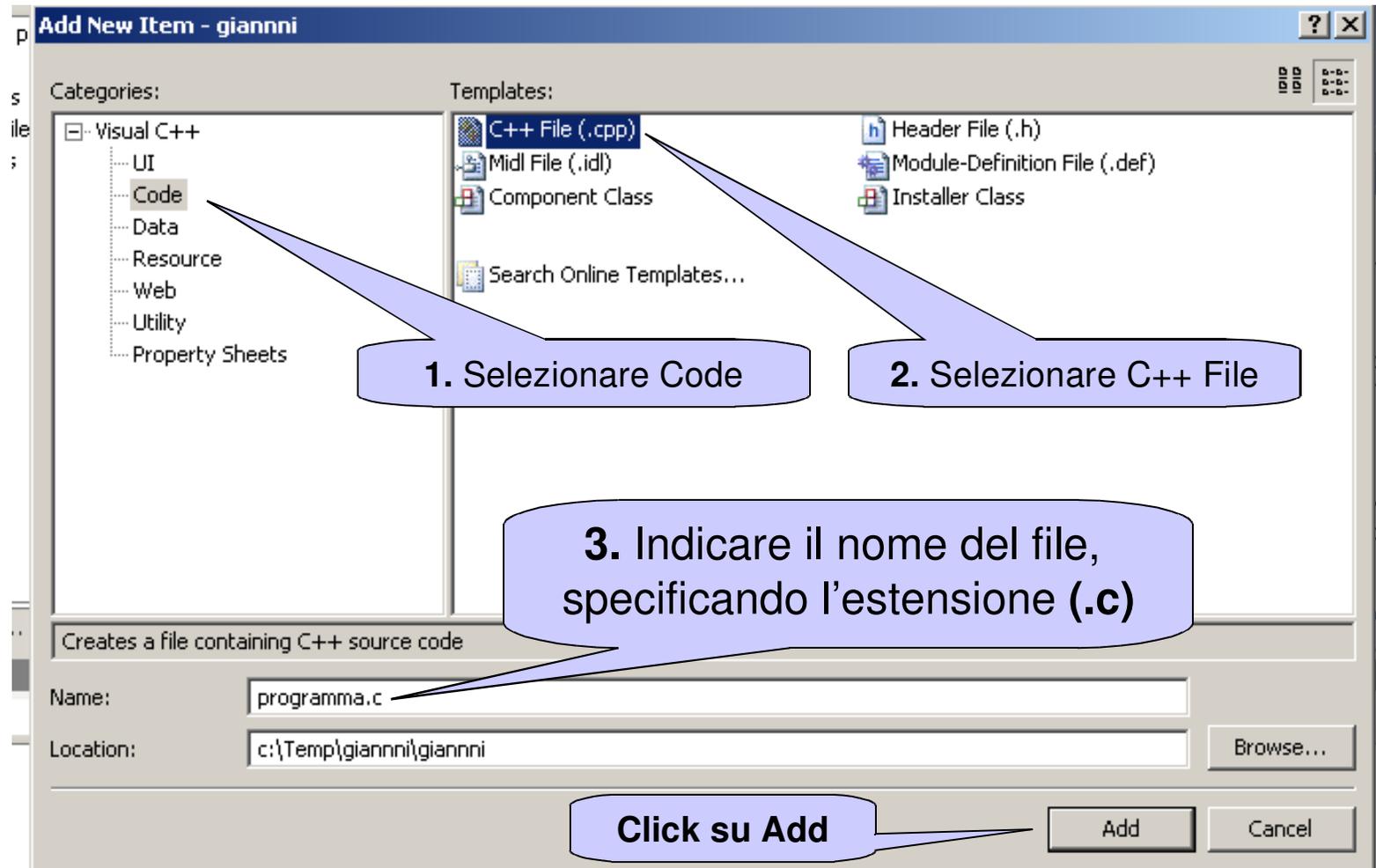


Aggiungere file al progetto

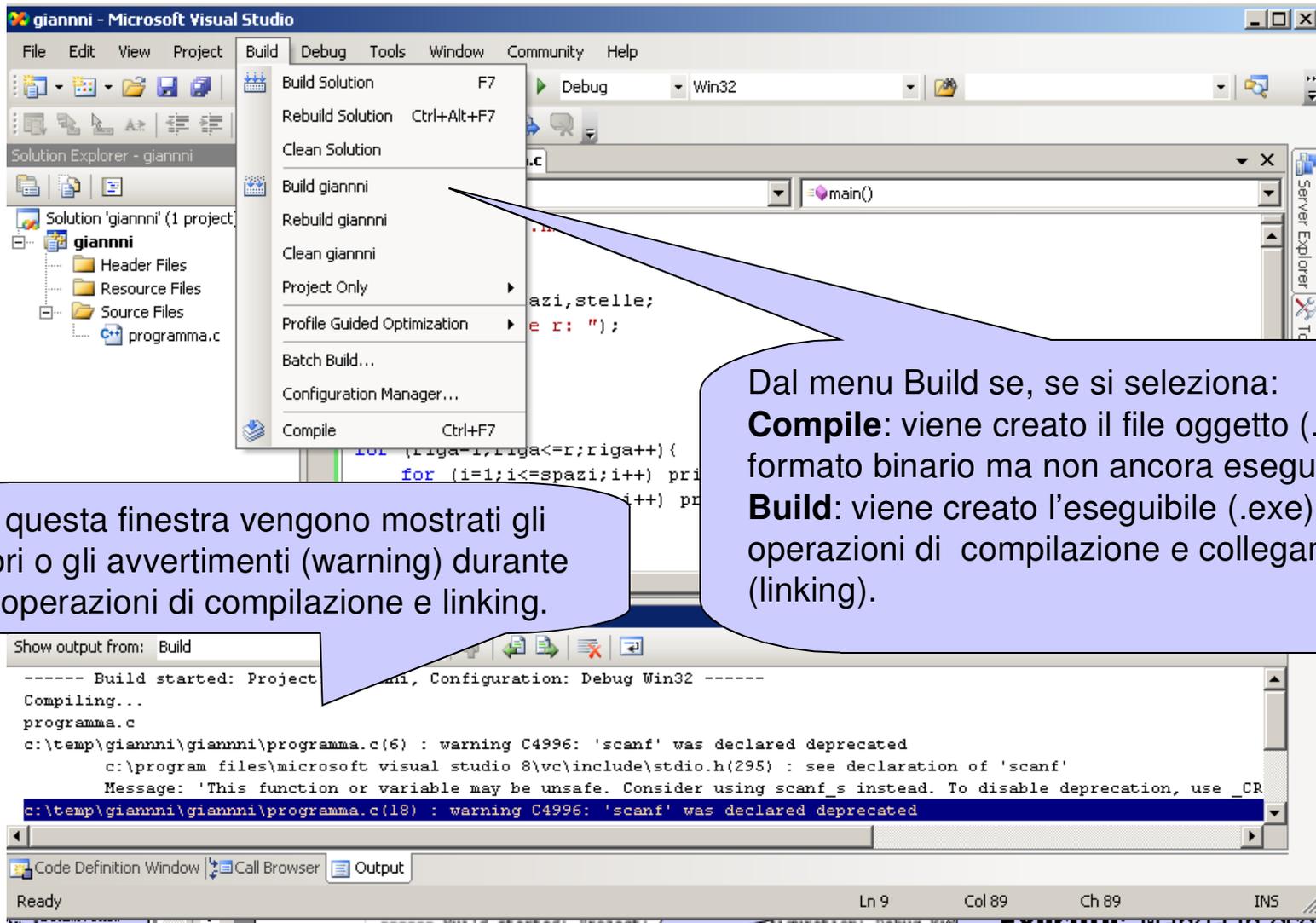


Si deve cliccare con il tasto destro sul nome del progetto, poi:
Add > New Item...

Aggiungere file al progetto



Compile, Build



In questa finestra vengono mostrati gli errori o gli avvertimenti (warning) durante le operazioni di compilazione e linking.

Dal menu Build se, se si seleziona:
Compile: viene creato il file oggetto (.obj), in formato binario ma non ancora eseguibile.
Build: viene creato l'eseguibile (.exe) Compie le operazioni di compilazione e collegamento (linking).

Compile, Build

gianni - Microsoft Visual Studio

File Edit View Project Build Debug Tools Window Community Help

Solution Explorer - Solution 'gianni' (1 project)

- gianni
 - Header Files
 - Resource Files
 - Source Files
 - programma.c

```
#include <stdio.h>

void main() {
    int i,r,riga,spazi,stelle;
    printf("Inserire r: ");
    scanf("%d",&r);

    spazi=r-1;
    stelle=1;
    for (riga=1;riga<=r;riga++){
        for (i=1;i<=spazi;i++) printf(" ");
        for (i=1;i<=stelle;i++) printf("*");
        printf("\n");
        spazi--;
        stelle++;
        stelle++;
    }
    scanf("%d",&r);
}
```

Per eseguire il programma click sulla freccina verde.

La console si chiude subito senza mostrarci il risultato? Inseriamo una scanf inutile alla fine! (Così aspetterà che scriviamo qualcosa)

Output

Show output from: Debug

```
'gianni.exe': Loaded 'C:\Temp\gianni\debug\gianni.exe', Symbols loaded.
'gianni.exe': Loaded 'C:\WINNT\system32\NTDLL.DLL', Cannot find or open a required DBG file.
'gianni.exe': Loaded 'C:\WINNT\system32\msvcr80d.dll', Symbols loaded.
'gianni.exe': Loaded 'C:\WINNT\system32\msvcrt.dll', No symbols loaded.
'gianni.exe': Loaded 'C:\WINNT\system32\KERNEL32.DLL', Cannot find or open a required DBG file.
The program '[640] gianni.exe: Native' has exited with code 0 (0x0).
```

Ready Ln 20 Col 1 Ch 1 INS

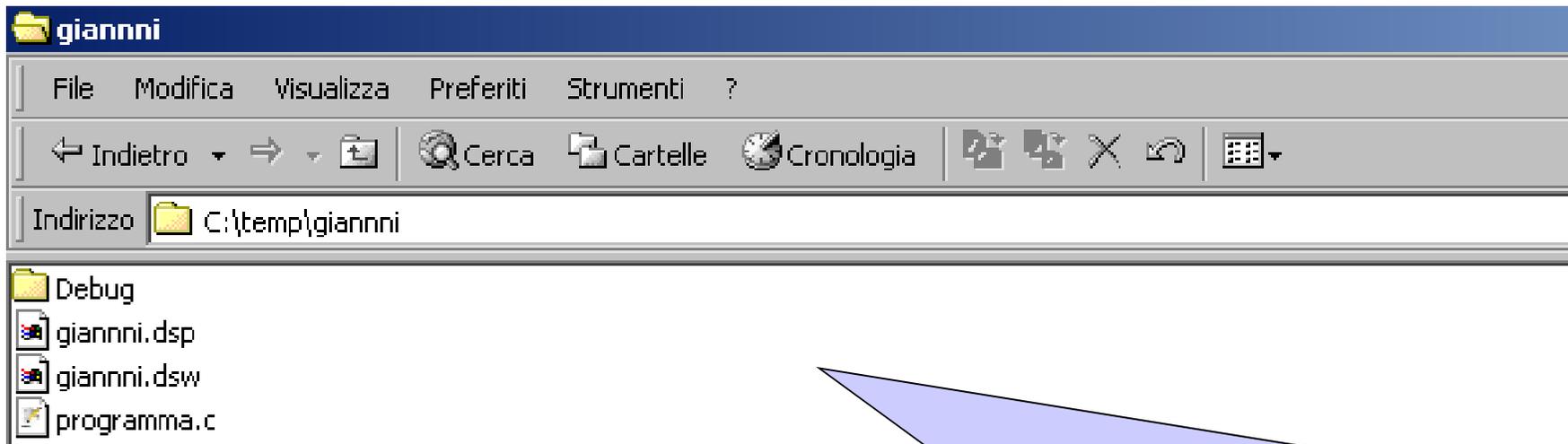
HELLO WORLD

- Il mio primo programma:

- Stampa a video il messaggio "Hello world!!!"
- Ricordarsi di includere `stdio.h` (sono le librerie per l'input/output)

```
#include <stdio.h>
main() {
    printf("Hello World!!! \n");
    scanf("\n"); /*solo per tenere
                 la finestra aperta*/
}
```

Uno sguardo al file system



Abbiamo creato il progetto sotto **c:\temp**. All'atto della creazione del progetto viene creata una cartella con il nome del progetto: **gianni**.

Nella cartella **c:\temp\gianni** è possibile osservare i file **.dsp** e **.dsw** usati da Visual Studio per memorizzare informazioni relative all'organizzazione del progetto.

Soprattutto, in questa cartella sono memorizzati anche i sorgenti, vediamo infatti il nostro **programma.c**. Nella cartella **Debug** durante l'operazione di building, vengono creati i file oggetto (**programma.obj**) e l'eseguibile (**gianni.exe**).

Lanciare l'eseguibile a riga di comando

- Aprire la finestra del prompt dei comandi:
 - **Start > Programs > Accessories > Command Prompt**
oppure
 - **Start > Run ... digitare cmd e OK**
- Entrare nella cartella dove risiede il file .exe
 - `cd c:\temp\<NOMEPROGETTO>\debug`
- Digitare il nome del file (eventualmente seguito da parametri) e INVIO.

COSTRUZIONE DI UN'APPLICAZIONE

Per costruire un'applicazione occorre:

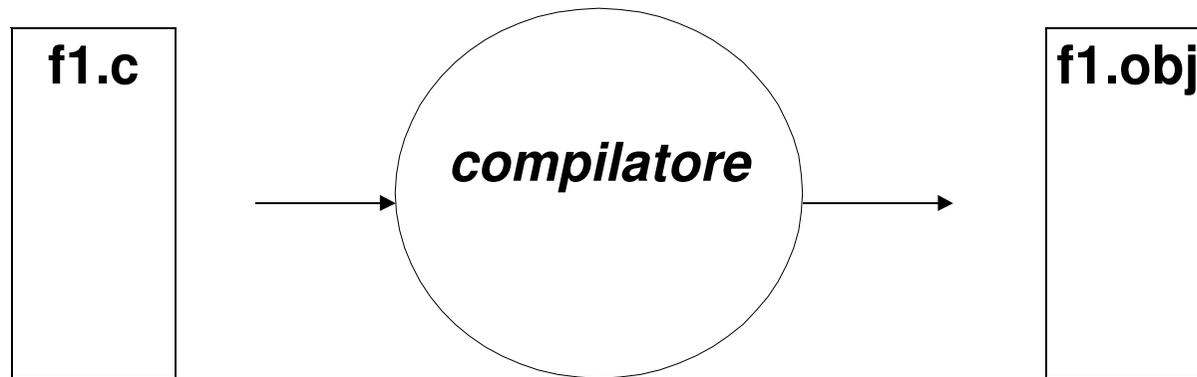
- **compilare il file (o / file se più d'uno) che contengono il testo del programma (file *sorgente*)**
Il risultato sono uno o più file *oggetto*.
- **collegare i file oggetto l'uno con l'altro e con le librerie di sistema.**

COMPILAZIONE DI UN'APPLICAZIONE

1) **Compilare il file (o / file se più d'uno) che contengono il testo del programma**

– **File sorgente:** estensione **.c**

– **File oggetto:** estensione **.o** o **.obj**

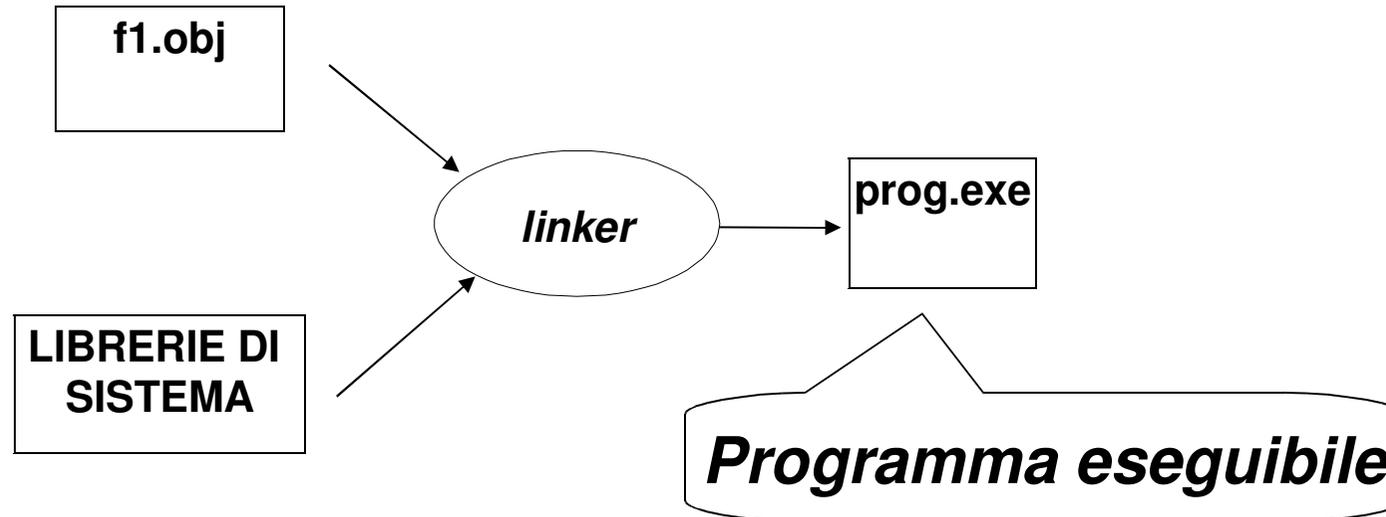


f1.obj: Una versione tradotta che però non è autonoma (e, quindi, non è direttamente eseguibile).

COLLEGAMENTO DI UN'APPLICAZIONE

2) Collegare il file (o *i* file) oggetto fra loro e con le librerie di sistema

- File *oggetto*: estensione **.o** o **.obj**
- File *eseguibile*: estensione **.exe** o nessuna



COLLEGAMENTO DI UN'APPLICAZIONE

LIBRERIE DI SISTEMA:

insieme di componenti software che consentono di interfacciarsi col sistema operativo, usare le risorse da esso gestite, e realizzare alcune "istruzioni complesse" del linguaggio.

AMBIENTI INTEGRATI

Oggi, gli *ambienti di lavoro integrati* automatizzano la procedura:

- **compilano i file sorgente (*se e quando necessario*)**
- **invocano il linker per costruire l'eseguibile**

ma per farlo devono sapere:

- ***quali file sorgente costituiscono l'applicazione***
- ***il nome dell'eseguibile da produrre.***

PROGETTI

È da queste esigenze che nasce il concetto di **PROGETTO**

- ***un contenitore concettuale (e fisico)***
- ***che elenca i file sorgente in cui l'applicazione è strutturata***
- ***ed eventualmente altre informazioni utili.***

Oggi, *tutti* gli ambienti di sviluppo integrati, *per qualunque linguaggio*, forniscono questo concetto e lo supportano con idonei strumenti.