INTRODUZIONE A VISUAL STUDIO

Cosa è Visual Studio

- Visual Studio è un ambiente di sviluppo integrato (Integrated development environment o IDE)
- Sviluppato da Microsoft
- Supporta numerosi linguaggi
  - C, C++, C#,
  - F#, Visual Basic .Net e ASP .Net,
  - …..

Usare Visual Studio

- Solution e Project
- Visual Studio e linguaggio C
- Visual Studio – schermata principale
- Aggiungere file a un progetto
- Compilare ed eseguire un programma
- Debug di un programma
- “Che cosa fare se...”
- “Creare un progetto per il C”

Soluzioni (solutions) e Progetti (projects)

- In Visual Studio, ogni programma si sviluppa come un “progetto”...
- Un progetto contiene tutte le informazioni utili/necessarie per realizzare il programma
  - Elenco dei file sorgenti che compongono quel programma
  - Opzioni particolari relative allo specifico progetto

Soluzioni (solutions) e Progetti (projects)

A volte un programma “usa”funzionalita offerte da un altro programma

In tal caso è utile avere due progetti separati (uno per ogni programma)...

... ma è utile anche raggruppare i programmi

- secondo criteri di utilizzo (il programma A usa il programma B)
- secondo criteri di affinità funzionali (i programmi A e B svolgono compiti molto simili)
- secondo criteri di composizione (i programmi A e B condividono lo stesso componente)
- ..

Soluzioni (solutions) e Progetti (projects)

- Una Solution è un insieme di progetti, raggruppati secondo qualche criterio o esigenza
- Una Solution è composta da:
  - uno o più progetti
  - opzioni particolari relative alla specifica solution
- Vantaggi:
  - Riusabilità dei singoli progetti
  - Modularità nella realizzazione della Solution

Visual Studio e il linguaggio C

- Visual Studio “a default” supporta il linguaggio C++, non direttamente il linguaggio C
- C++ è sintatticamente derivato da C…
  - È possibile usare Visual Studio per realizzare programmi in C ... però è necessario specificare nelle opzioni di progetto che si sta scrivendo un programma in linguaggio C!!! Altrimenti:
    - Il compilatore non rileva alcuni errori
    - Può segnalare errori inconsistenti con le regole del linguaggio C

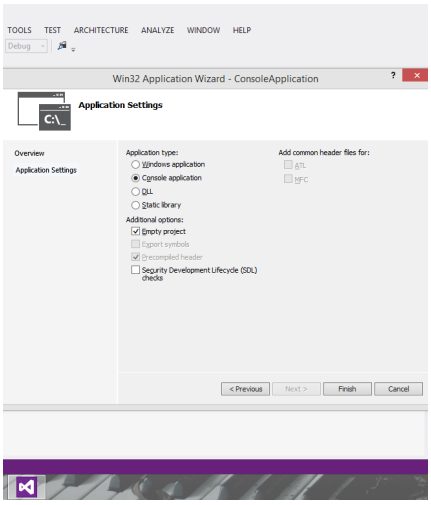
Visual Studio e il linguaggio C

Per creare progetti C,

Creare un progetto nuovo

1. a) Si specifica di voler creare un progetto “Console Application”
2. b) Si specificano le opzioni particolari per il linguaggio C

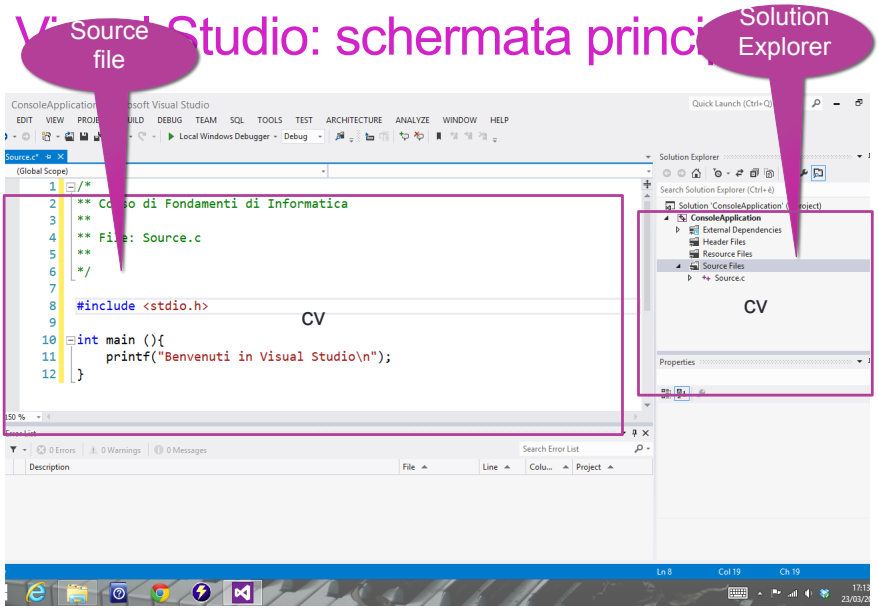
Il progetto può essere creato in una nuova solizione o aggiunto in una soluzione già esistente



Configuro il progetto per il C

- Èsufficiente aggiungere due configurazioni particolari:
- ***1. Specificarel'uso del solo linguaggioC usando come estensione .c (e non .cpp)***: in questo modo Visual Studio verificherà automaticamente che il programma sia effettivamente scritto in C e non in C++...
- ***2. Specificare di segnalare come warning*** l'uso di alcune **funzioni standard** del C il cui uso può risultare critico (non selezionando SDL)

Visual Studio: schermata princi



Visual Studio: schermata principale

Solution Explorer

- Mostra l'elenco dei progetti e dei file appartenenti ad ogni progetto. Per
  - aprire un file, basta fare "doppio click" su di esso... • **Source files**
- Mostra i file aperti, ogni file in un tab separato • **Additional Windows**
- Mostrano alcune finstre ausiliarie molto importanti, quali:
  - "Output": mostra i messaggi di errore o di successo forniti dal framework
  - "Error List": elenco degli errori e dei warning rilevati in fase di compilazione. Cliccando su un errore, viene aperto il file corrispondente, e il cursore si posiziona nel luogo dove il compilatore presume ci sia l'errore...

Compila (build) e Debug

Contengono i pulsanti per compilare e per fare il debugging di un programma

Visual Studio: Error List window

Contiene la ***lista degli errori e dei warning*** rilevati dal compilatore

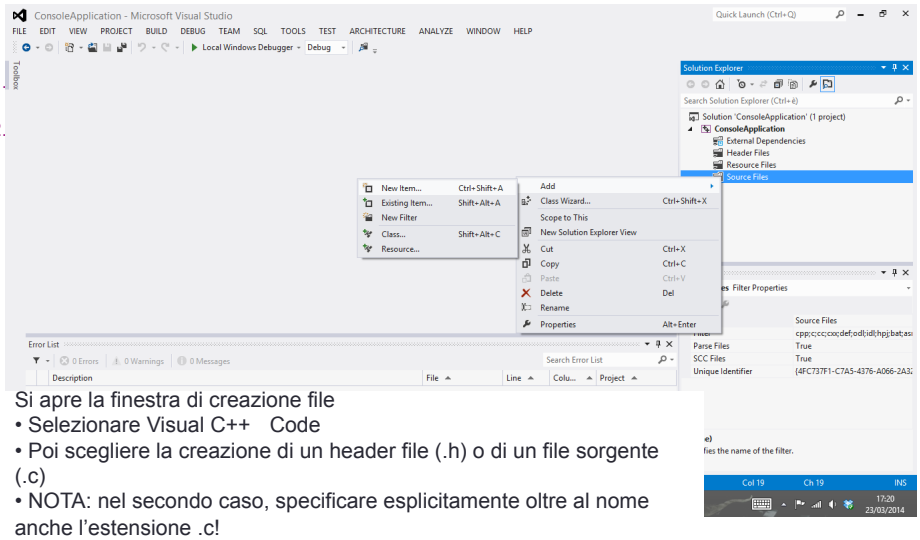
Gli **errori** sono situazioni gravi, che rendono impossibile compilare il programma

I **warning** sono situazioni in cui qualcosa di strano è stato rilevato dal compilatore, che però riesce a compilare comunque... ma spesso sono sintomi di errori non trascurabili

Un programma ben fatto, al momento della compilazione:

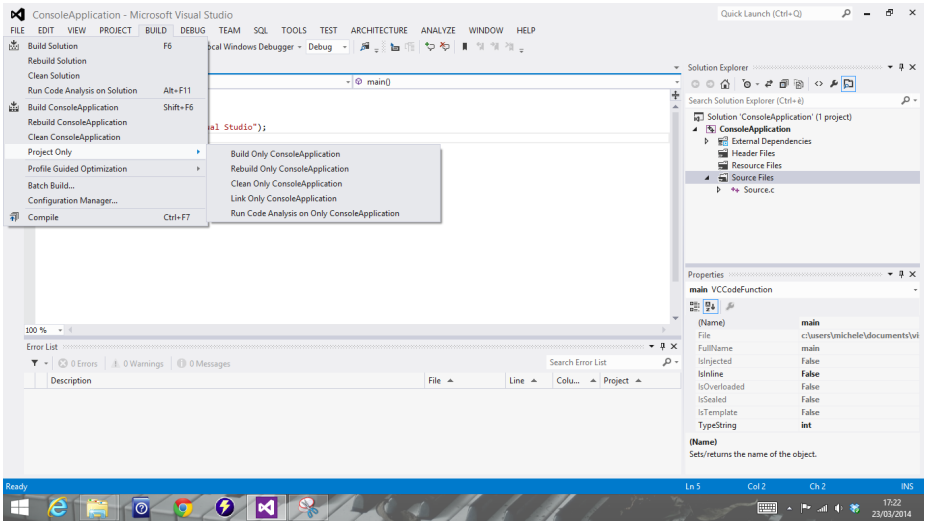
- **Non contiene errori**
- **Non genera warning**

Aggiungere files ad un progetto

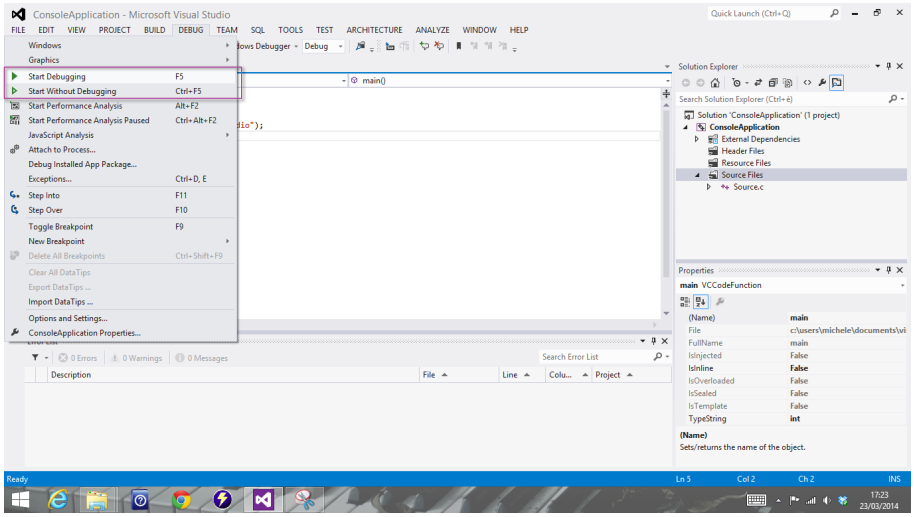


Si apre la finestra di creazione file
- Selezionare Visual C++ Code
- Poi scegliere la creazione di un header file (.h) o di un file sorgente (.c)
- NOTA: nel secondo caso, specificare esplicitamente oltre al nome anche l'estensione .c!

Compilazione/Linking



Esecuzione di un programma



Debug di un programma

- Col termine “Debug” si intende una fase di sviluppo del software, nella quale si cerca di eliminare gli errori dal programma
- Due tipi di errori:
  - **Errori sintattici**, rilevati sempre dal compilatore in fase di compilazione

    Errori semantici, difficilmente rilevabili Esempio: un programma deve eseguire la somma di due numeri,
- Esempio: un programma deve eseguire la somma di due numeri, ma il programmatore in un momento di distrazione ha usato il simbolo di operazione “-” invece del simbolo “+”
- **CONSEGUENZE: il programma è sintatticamente corretto, ma non esegue ciò che è stato richiesto**

Debug di un programma

- Il programmatore deve essere in grado, per ogni istruzione del proprio programma, di prevedere che cosa farà tale istruzione, cioè
- **Il programmatore deve conoscere in anticipo gli effetti derivanti dall'eseguire una certa istruzione**

IDEA: per ogni istruzione del programma:

1. Calcolo quali siano gli effetti nell'eseguire l'istruzione
2. Eseguo tale istruzione
3. Verifico che gli effetti siano effettivamente ciò che mi aspettavo
4. Se la verifica fallisce, ho trovato un errore ☺!

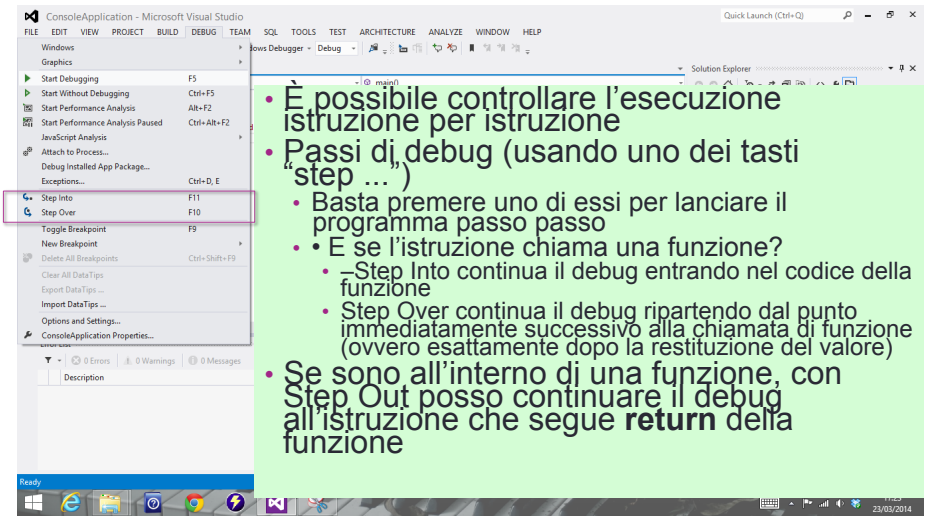
Uso del debug

- L'ambiente di sviluppo ci mette a disposizione una serie di funzionalità per:
  - Eseguire passo passo ogni istruzione
  - Controllare lo "stato" del nostro programma
    - Visualizzare il contenuto delle variabili (monitoraggio)
    - Visualizzare lo ***stack delle chiamate a funzione*** *(imparerete presto che cosa significa...)*
    - –...

Lancio di un programma e debug

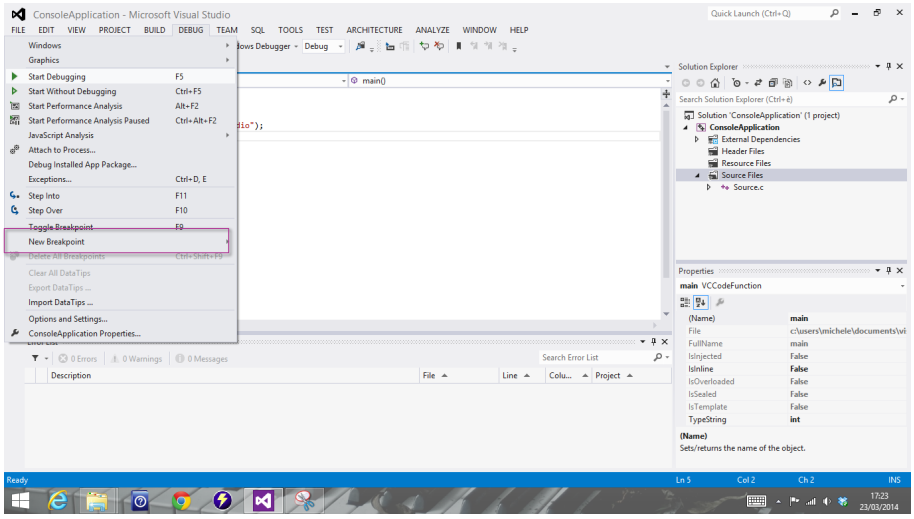
- Selezionare “Start debug” dal menù “Debug”
  - Il programma viene lanciato in modalità debug
  - Da un punto di vista dell’esecuzione non cambia niente...
  - ...ma vi dà la possibilità di andare a controllare il vostro codice istruzione per istruzione

Debug

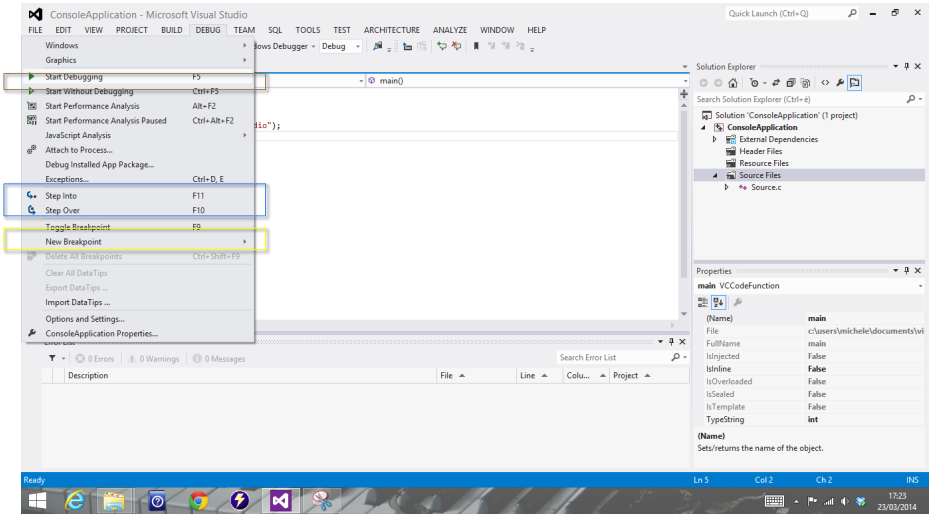


- È possibile controllare l’esecuzione istruzione per istruzione
- Passi di debug (usando uno dei tasti “step ...”)
  - Basta premere uno di essi per lanciare il programma passo passo
  - • E se l’istruzione chiama una funzione?
    - –Step Into continua il debug entrando nel codice della funzione
    - Step Over continua il debug ripartendo dal punto immediatamente successivo alla chiamata di funzione (ovvero esattamente dopo la restituzione del valore)
- Se sono all’interno di una funzione, con Step Out posso continuare il debug all’istruzione che segue **return** della funzione

Breakpoints (1)



Breakpoints



Monitoraggio delle variabili

Tre finestre di monitoraggio delle variabili

Auto

- Visualizza il contenuto delle variabili definite ***all'internodello scope corrente*** (e anche il valore di ritorno all'uscita da una funzione)

Local

- Visualizza il contenuto delle ***variabili"locali",ovvero tutte quelle visibili all'interno della funzione corrente*** (nota: in caso di scope innestati con variabili con lo stesso nome, compaiono ripetizioni)

Watch

- Permette di inserire il ***nome della variabile da monitorare*** (attenzione agli scope)
- È possibile anche monitorare espressioni(es:a+b)

Finestra Call Stack

- Permette di visualizzare lo stack delle chiamate a funzione – Alla chiamata di una funzione viene aggiunta una riga che mostra il ***valore dei parametri attuali***
- All'uscita di una funzione rimozione della riga(incima)
- È possibile selezionare una qualsiasi delle righe, e le finestre di monitoraggio delle variabili recuperano lo stato corrispondente
- • Provare con funzioni ricorsive!
- Call stack del fattoriale ricorsivo...

Cosa fare se ….

... non compare il Solution Explorer

1. Menù “View”
2. Selezionare la voce “Solution Explorer”

... non compare la finestra “Output” in basso

1. Menù “View”
2. Selezionare la voce “Output”

... non compare la finestra “Error List” in basso

1. Menù “View”
2. Selezionare la voce “Error List”

... non compare la “Build Palette”

1. Cliccare con il tasto di destra del mouse un punto
2. qualunque sulla barra dei bottoni o dei menu
3. Selezionare la voce “Build”

... non compare la “Debug Palette”

1. Cliccare con il tasto di destra del mouse un punto qualunque sulla barra dei bottoni o dei menu
2. Selezionare la voce “Debug”