



IDE – Ambiente di sviluppo integrato

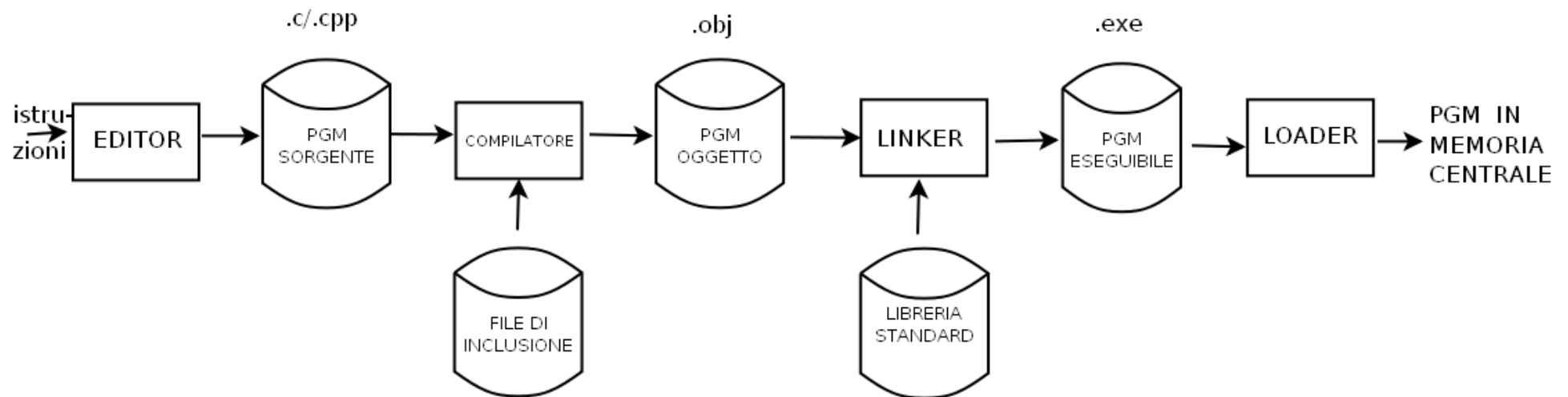


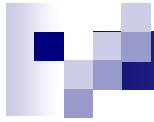
IDE - Integrated Development Environment

- Ambiente di sviluppo integrato, comprende una varietà di strumenti coordinati per supportare il processo di sviluppo dei programmi (creazione, traduzione, esecuzione, test, ...), tra cui:
 - ☐ editor
 - ☐ compilatore
 - ☐ linker
 - ☐ debugger
 - ☐ help
 - ☐

- Dev-C++: IDE per programmi C/C++
 - ☐ free
 - ☐ tra i più semplici a disposizione
(ma noi ne useremo comunque solo una piccola parte!)

Fasi e file coinvolti nella costruzione di un pgm C/C++ eseguibile a partire dal sorgente





Alcune definizioni

- Un **editor** è un pgm che permette di creare, modificare, salvare e stampare un pgm sorgente.
- Un **pgm sorgente** è un pgm scritto in un linguaggio di programmazione. (E' un file di testo).
- Un **compilatore** è un programma che, dopo aver svolto un'analisi sintattica, traduce una serie di istruzioni scritte in un determinato linguaggio di programmazione (codice sorgente) in istruzioni di un altro linguaggio (codice oggetto).



Alcune definizioni

- Il **linker** è un programma che effettua il collegamento tra il programma oggetto, cioè la traduzione del codice sorgente in linguaggio macchina, e le librerie del linguaggio necessarie per l'esecuzione del programma creando il pgm eseguibile.
- Una **libreria** (software) è un insieme di funzioni (in formato compilato) predisposte per essere collegate ad un programma.



Alcune definizioni

- Il **debugger** è un programma progettato per l'analisi e l'eliminazione dei bug (ovvero errori di programmazione interni al programma) presenti in un programma. Esegue la trace di un pgm in modo automatico.

- **Tipi di errori:**
 - ☐ di compilazione (sintattici) (es. manca il punto e virgola al termine di un'istruzione, ...)
 - ☐ di esecuzione – run time (es. divisione per zero, ...)
 - ☐ logici: i risultati del pgm sono diversi da quelli attesi

- **Help** = pgm di aiuto: online, contestuale, ipertestuale.



Il compilatore: warning e error

■ ***warning***

- ☐ messaggi di avvertimento (attenzione!)
- ☐ non interrompono la compilazione
- ☐ avvisano della (possibile) presenza di irregolarità nel codice
- ☐ inibire i messaggi di warning (sconsigliato)
- ☐ settare al massimo il livello di warning (consigliato)

■ ***error***

- ☐ interrompono la compilazione
- ☐ indicano errori che devono essere corretti

- Nota: anche il linker può segnalare errori.



IDE Dev-C++

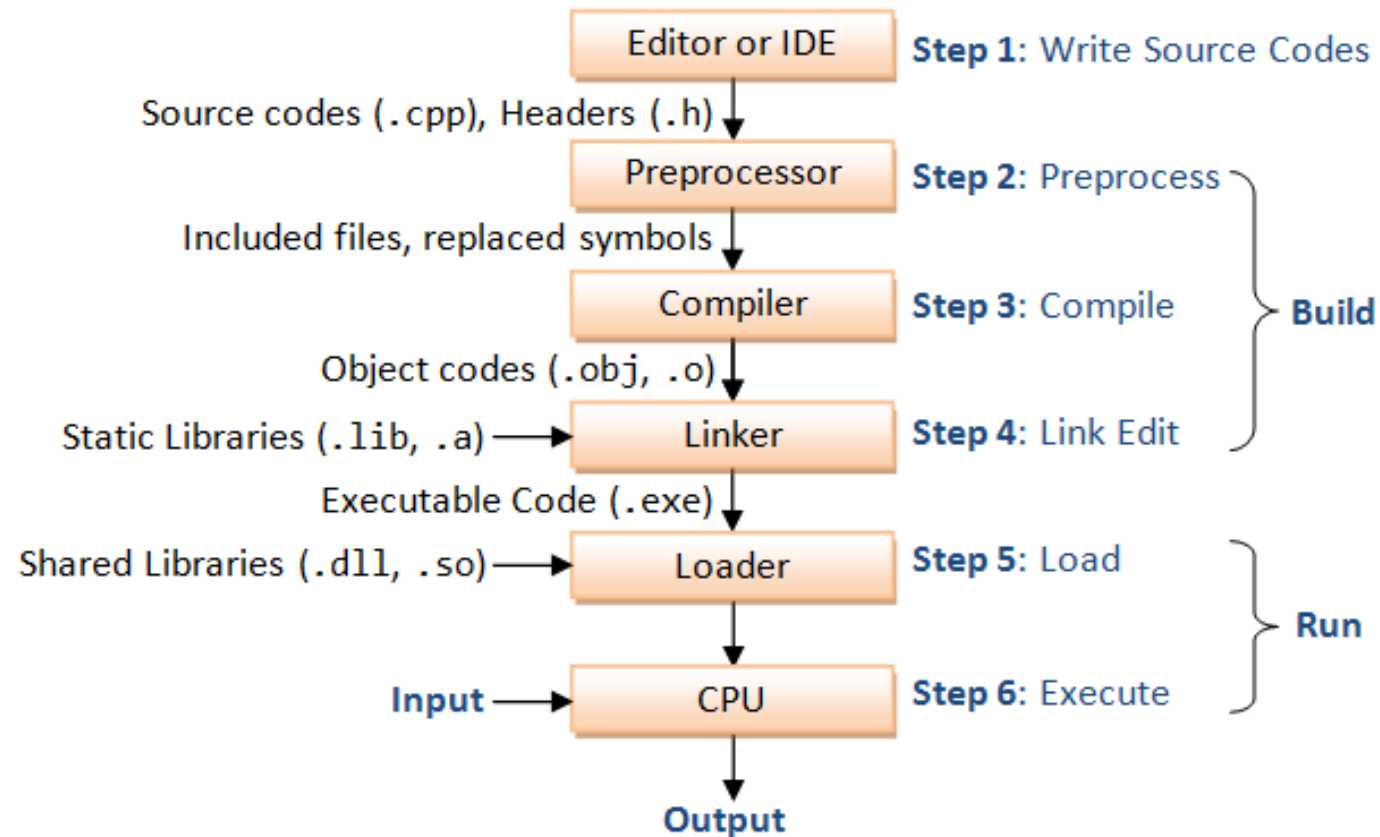
- Dev-C++ lavora su ***Progetti***, ossia delle cartelle in cui si va inserire tutto il necessario per poter creare un programma C++ funzionante.
- Noi cominceremo a programmare partendo da un singolo file:
 - File | Nuovo | File Sorgente
- Utilizzate sempre nomi significativi, senza spazi e caratteri speciali. Ricordatevi di aggiungere alla fine del nome l'estensione .cpp o .c.



IDE Dev-C++

- Terminata la scrittura del programma, si può compilare e linkare il programma, cioè tradurre da programma scritto in linguaggio di alto livello (il C++) in linguaggio macchina, premendo il pulsante “Compila” (Compiler + Linker). Una volta compilato e linkato, è possibile eseguirlo tutte le volte che si vuole premendo sul pulsante “Run” .
- RICORDATE Ogni qualvolta che modificate il vostro programma C++, il programma DEVE essere compilato di nuovo attraverso il pulsante “Compila”, altrimenti le modifiche che farete NON si potranno vedere nelle esecuzioni successive alla modifica.
- Per comodità, Dev-C++ fornisce il pulsante “Compila & Esegui” , che permette di ricompilare tutte le volte prima di eseguire il programma.

IDE Dev-C++





Come fare il debug di un pgm

■ Per adesso

- ☐ Inizia sempre dall'algoritmo (spesso è sufficiente una descrizione ad alto livello) scritto su carta.
- ☐ Esegui una rapida trace su carta.
- ☐ Scrittura del pgm sorgente.
- ☐ Compilazione e correzione errori sintattici
- ☐ Esecuzione
- ☐ Correzione errori logici con inserimento nel pgm sorgente di istruzioni di visualizzazione di variabili e/o stringhe opportune.
- ☐ Disegno e trace su carta.

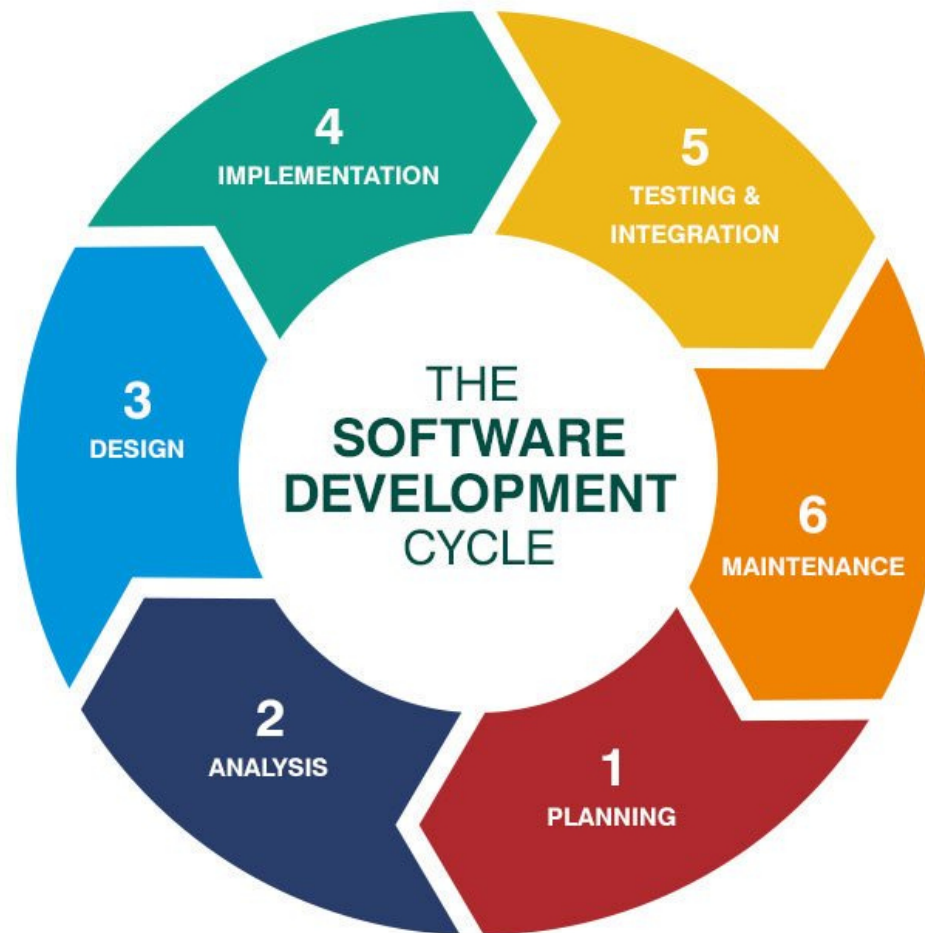
■ Prossimamente

- ☐ Tutto quanto detto nella colonna di sinistra vale ancora.
- ☐ Utilizzo del debug messo a disposizione dall'IDE.

NOTA BENE Per fare un buon test del pgm è necessario aver preparato un buon piano di test.

Ricorda: se nella fase di testing di un pgm non hai trovato errori ... non hai fatto un buon test!

Fasi del ciclo di vita del software





Condizione in assenza di modello

