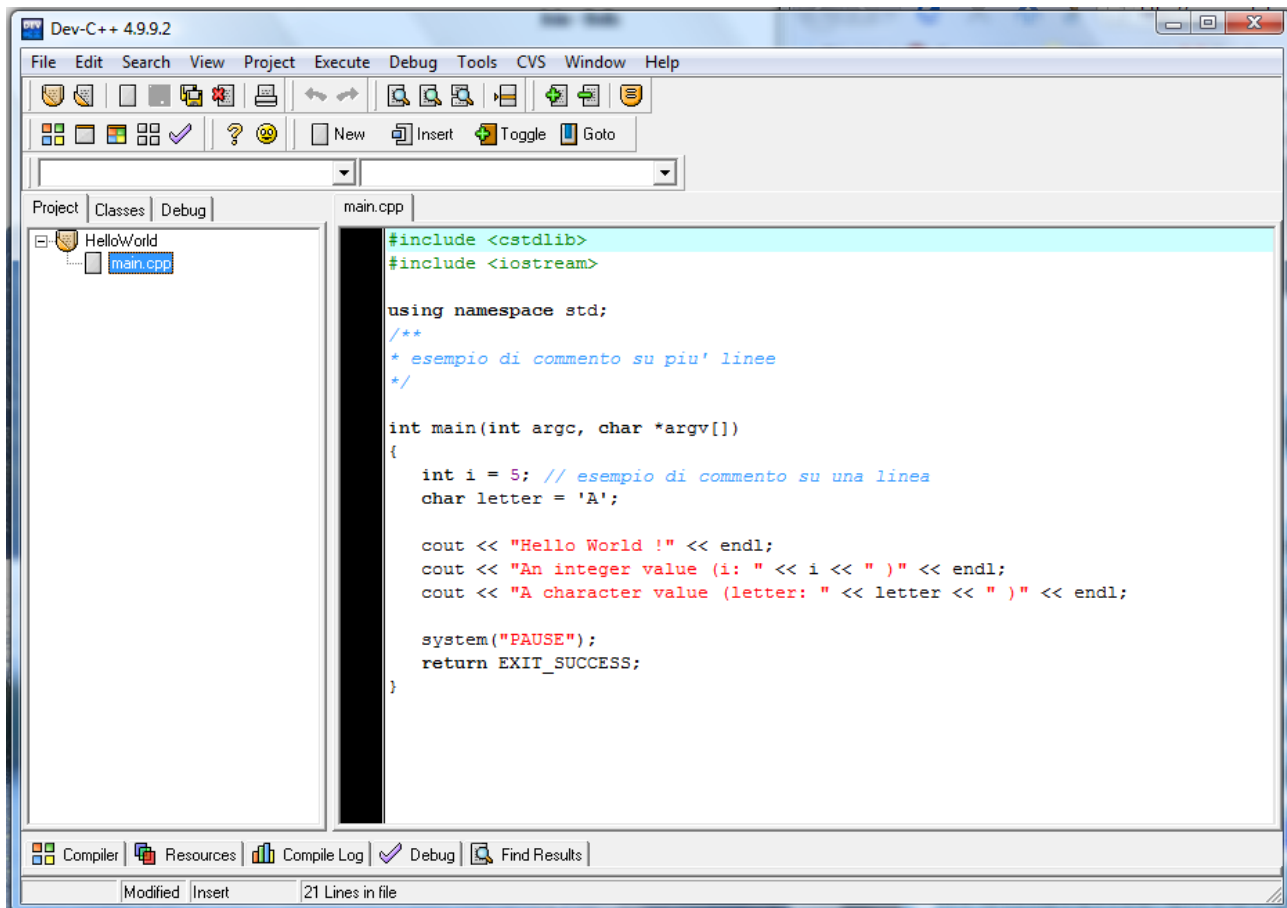


Bloodshed Dev-C++

- Bloodshed Dev-C++ è l'IDE usato durante le esercitazioni/laboratorio

– IDE = Integrated Development Environment



Dev-C++ - Installazione

- Con sistema operativo **Windows**, potete installare Dev-C++ sul vostro PC (consiglio: in lingua inglese)

– <http://orwelldevcpp.blogspot.it/>

Nella pagina al link sopra indicato, tra le diverse opzioni scegliete di eseguire il download e l'installazione indicata come segue:

The **setup** which includes **TDM-GCC 4.9.2** (32bit and 64bit) can be downloaded here (47MB)

– è consigliato usare il percorso di installazione seguente
C:\Program Files (x86)\Dev-Cpp

– se avete problemi installando l'ambiente di sviluppo su una versione precedente del sistema operativo: Windows Vista, Windows 7, Windows 8

- provate ad abilitare la retro-compatibilità (click col tasto destro sull'eseguibile, opzione retro-compatibilità)

- provate a entrare in **Tools | Compiler options**

– Nella scheda **Directories** -- sottoscheda **Binaries**, aggiungete **C:\Program Files (x86)\Dev-Cpp\libexec\gcc\mingw32\3.4.2**

– Nella scheda **Programs**, aggiungete come prefisso **C:\Program Files (x86)\Dev-Cpp\bin** a tutti i programmi elencati

Dev-C++ - Guida rapida

- Create una nuova cartella sull'hard drive: C:\NomeStudente\
• Create una nuova cartella sull'hard drive: C:\NomeStudente\Esempio1\
• Eseguire l'IDE Dev-C++

- Creare un nuovo progetto
 - Selezionate: *New | Project | Console Application*
Name: HelloWorld, C++ project
 - Right-click sul Tab *main.cpp* scegliere *close*
 - Selezionate: *New | New File*
salvate il nuovo file con estensione *.cpp (es: Hello.cpp)*
 - Ricopiate il seguente codice sorgente

```
#include <cstdlib>
#include <iostream>

using namespace std;
/*
   esempio di commento su piu' linee
*/

int main(int argc, char *argv[])
{
    int i = 5; // esempio di commento su una linea
    char letter = 'A';

    cout << "Hello World !" << endl;
    cout << "An integer value (i: " << i << ")" << endl;
    cout << "A character value (letter: " << letter << ")" << endl;

    system("PAUSE"); // COMANDO NON PORTABILE! solo per Win
    return EXIT_SUCCESS;
}
```

- Compilare ed eseguire il programma
 - selezionate *Execute | Compile & Run* (oppure F9)
 - in assenza di errori Dev-C++ compila il programma producendo l'eseguibile **.exe** e lo esegue in una finestra DOS

Note:

In ogni progetto **deve** esserci **1 solo** file sorgente che contiene il `...main(...)`

I nomi delle *directory* (cartelle) e i nomi dei *file*

- devono **sempre** iniziare con un carattere alfanumerico
- **non devono** contenere **spazi** o **caratteri di punteggiatura**

Sviluppo su MAC o LINUX

- Bloodshed Dev-C++ è disponibile **solo** per ambiente Windows
- In ambiente Mac-OS si consiglia l'uso dell'ambiente di sviluppo: Xcode IDE
 - se non è già presente nel PC, dovete scaricarlo a
 - <http://developer.apple.com/tools/xcode/>
- PER TUTTI I SISTEMI OPERATIVI un buon IDE e': "Code::Blocks"
(oppure per Linux, **gcc** a linea di comando!)

Esercizio di introduzione all'uso del compilatore DevC++

Il codice sorgente dei programmi elencati di seguito ha l'obiettivo di riepilogare la struttura generica di un programma C++:

- primitive base di I/O,
- dichiarazione e inizializzazione di variabili,
- uso di costrutti condizionali e/o iterativi con la relativa sintassi.

Compilare e correggere gli eventuali errori presenti nei sorgenti (*warnings* compresi – imparare a distinguere tra errori di sintassi, errori del *linker*, errori logici).

Si consiglia di leggerli accuratamente prima di trascriverli con l'*editor* messo a disposizione dall'IDE del DevC++ per sviluppare le vostre capacità di analisi del codice.

Le soluzioni mostrate sono solo una traccia e contengono degli errori **appositamente inseriti.**

somma.cpp	Esegue e visualizza a video il risultato della somma di 2 numeri inseriti da un utente
sums.cpp	Esegue la somma di più numeri con terminatore zero (quando l'utente inserisce la cifra 0 "zero" il programma termina e restituisce all'utente la somma)
massimo.cpp	Esegue e visualizza a video il risultato del massimo tra 2 numeri inseriti da un utente
max.cpp	Esegue il massimo di più numeri con terminatore zero (quando l'utente inserisce la cifra 0 "zero" il programma termina e restituisce all'utente il valore massimo inserito)

(1) Sorgente: somma.cpp

```
#include <iostream> // inclusione della libreria standard
#include <cstdlib> // per la funzione system(...)
using namespace std;

void main(int argc, char *argv[])
{
    int num1, num2, sum;
    sum = 0;

    cout << "\nSomma di due numeri\n\n";
    cout << "Introdurre i due numeri da sommare" << endl << endl;
    cout << "Primo numero : ";
    cin << num1;
    cout << "\nSecondo numero : ";
    cin >> num2;
    sum = num1 + num2;
    cout << "\n\nLa somma vale " << sum << "\n\n";

    system("PAUSE");
    return EXIT_SUCCESS;
}
```

(2) Sorgente: sums.cpp

```
#include <cstdlib> // per la funzione system(...)

int main(int argc, char *argv[])
{
    int num, sum;

    sum = 0;
    num = 1;
    while (num != 0)
    {
        cout << "\n Introdurre un numero (zero per terminare): ";
        cin >> num;
        sum = sum + num;
    }
    cout << "\nLa somma vale " << sum << endl << endl;

    system("PAUSE");
    return EXIT_SUCCESS;
}
```

(3) Sorgente: massimo.cpp

```
#include <iostream> // inclusione della libreria standard
#include <cstdlib> // per la funzione system(...)

int main(int argc, char *argv[])
{
    int num1, num2;

    cout << "\nMassimo tra due numeri\n\n";
    cout << "Introdurre i due numeri da confrontare \n\n";
    cout << "Primo numero : ";
    cin >> num1;
    cout << "\nSecondo numero : ";
    cin >> num2;
    while (num1 < num2);
        cout << "\n" << num1 << " e' maggiore di " << num2 << "\n\n";
    else {
        cout << "\n" << num1 << " e' minore di (o uguale a);
        cout << num2 << "\n\n";
    }

    system("PAUSE");
    return EXIT_SUCCESS;
}
```

(4) Sorgente: max.cpp

```
#include <iostream> // inclusione della libreria standard
#include <cstdlib> // per la funzione system(...)

int main(int argc, char *argv[])
{
    int num, max;

    max = 0;
    num = 1;
    while (num != 0);
    {
        std::cout << "Introdurre un numero (zero per terminare):";
        std::cin >> num;
        if (num > max) max = num;
    }
    std::cout << "\nil massimo vale " << max << "\n";

    system("PAUSE");
    return EXIT_SUCCESS;
}
```