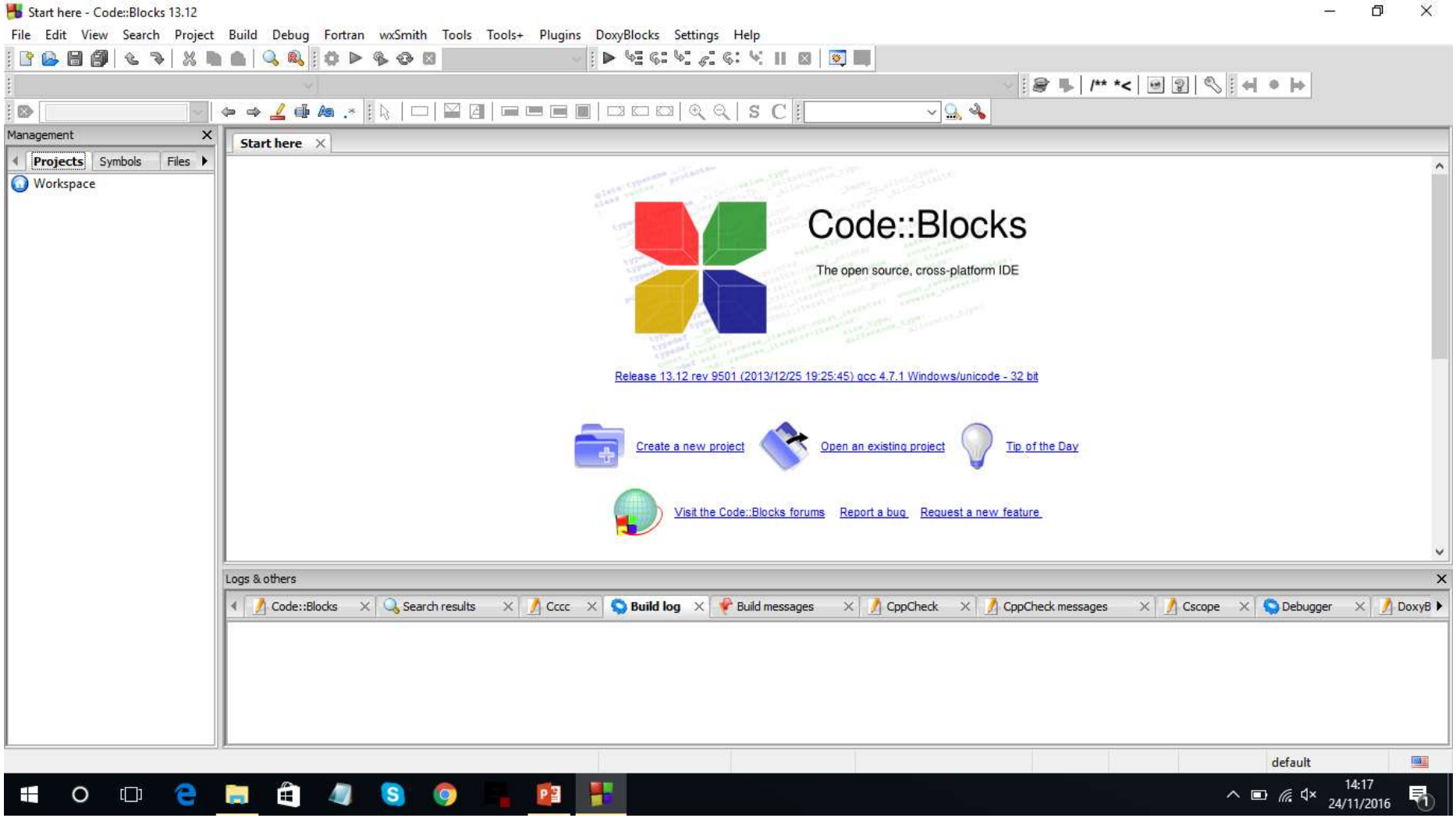
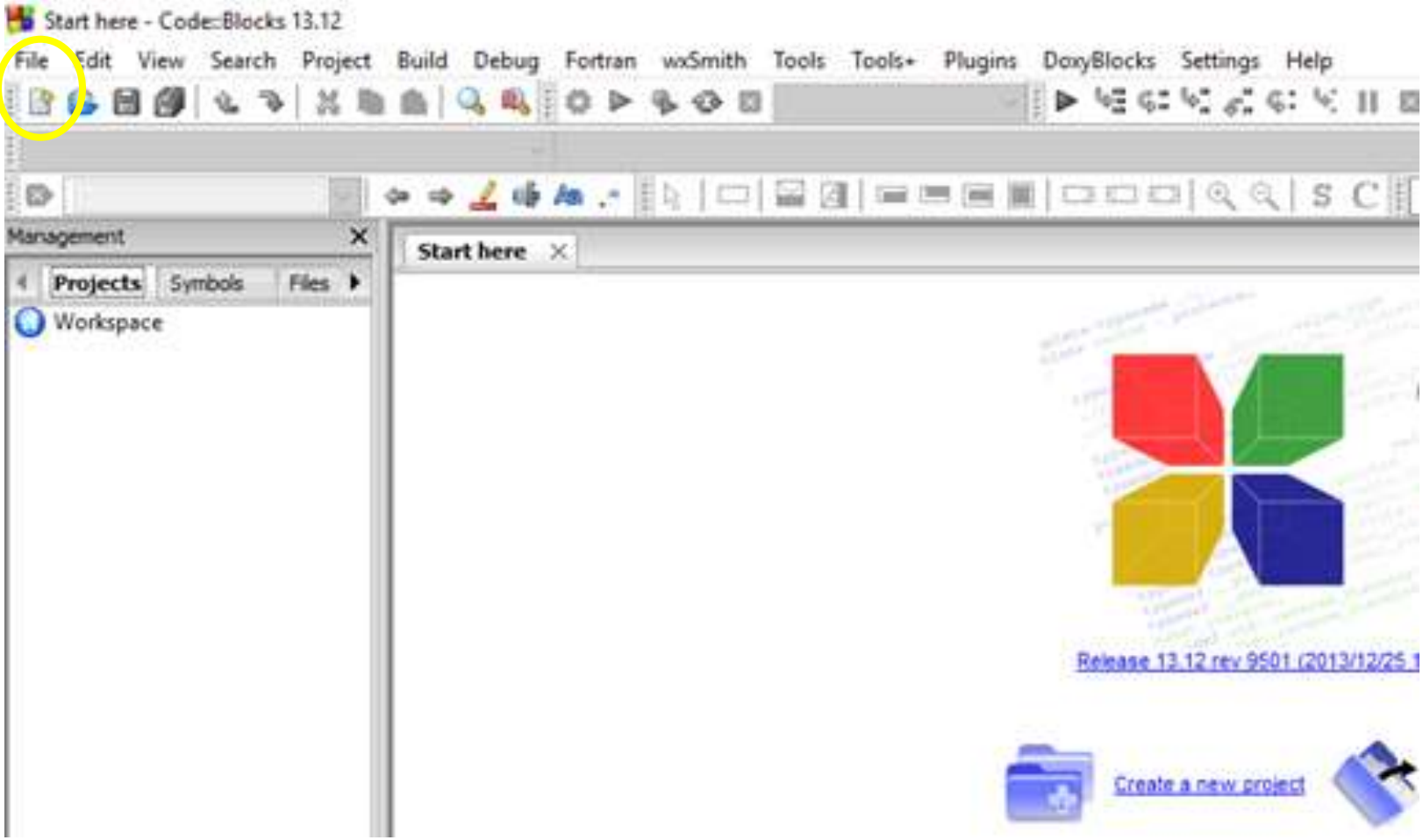
L'AMBIENTE CODE BLOCKS E L'IO


Start here - Code::Blocks 13.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Management
Projects Symbols Files
Workspace
Start here
Code::Blocks
The open source, cross-platform IDE
Release 13.12 rev 9501 (2013/12/25 19:25:45) gcc 4.7.1 Windows/unicode - 32 bit
Create a new project
Open an existing project
Tip of the Day
Visit the Code::Blocks forums
Report a bug
Request a new feature
Logs & others
Code::Blocks
Search results
Cccc
Build log
Build messages
CppCheck
CppCheck messages
Cscope
Debugger
DoxyB
default
14:17
24/11/2016


Start here - Code::Blocks 13.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Management
Projects Symbols Files
Workspace
Start here
Release 13.12 rev 9501 (2013/12/25
Create a new project

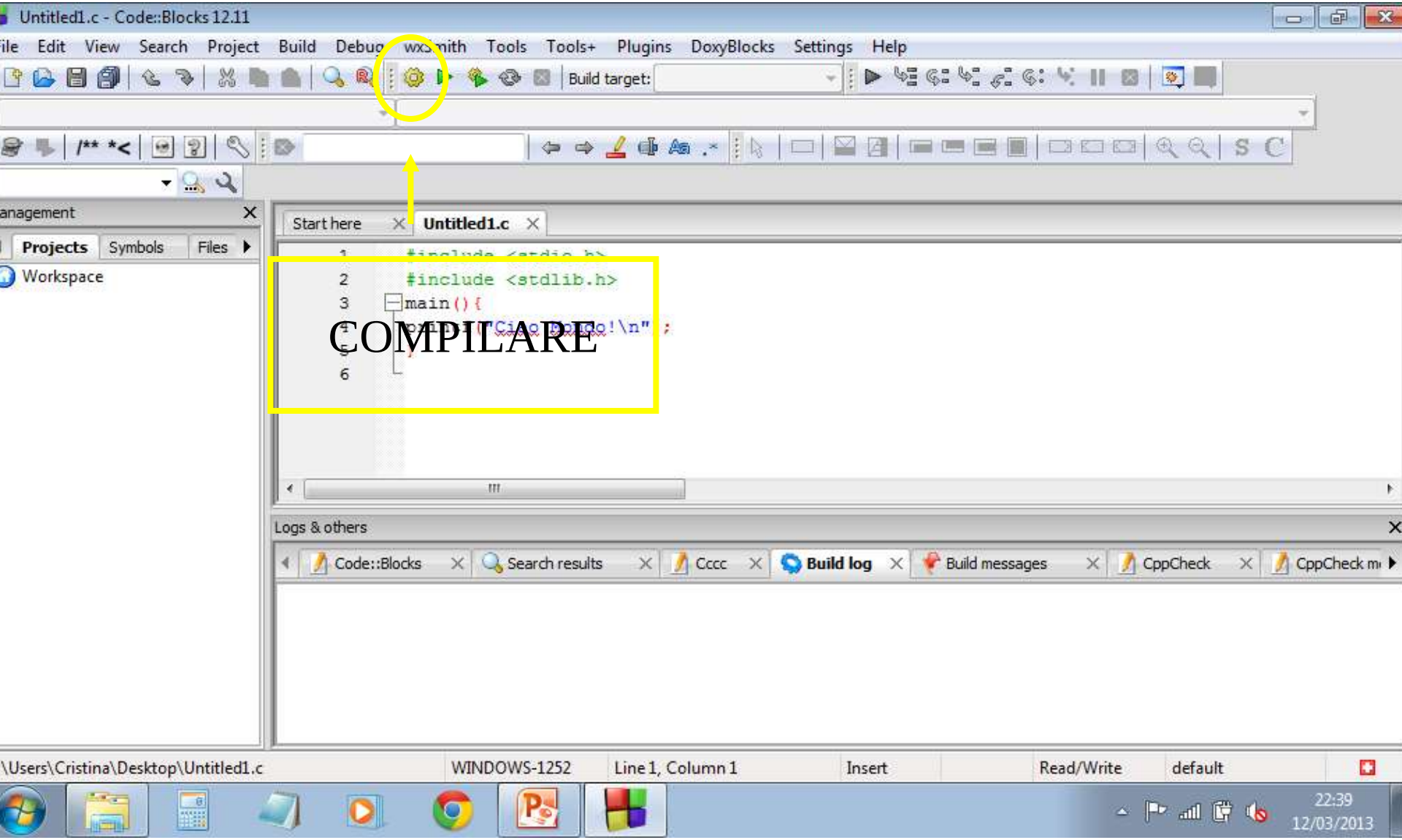
Il primo programma in C++

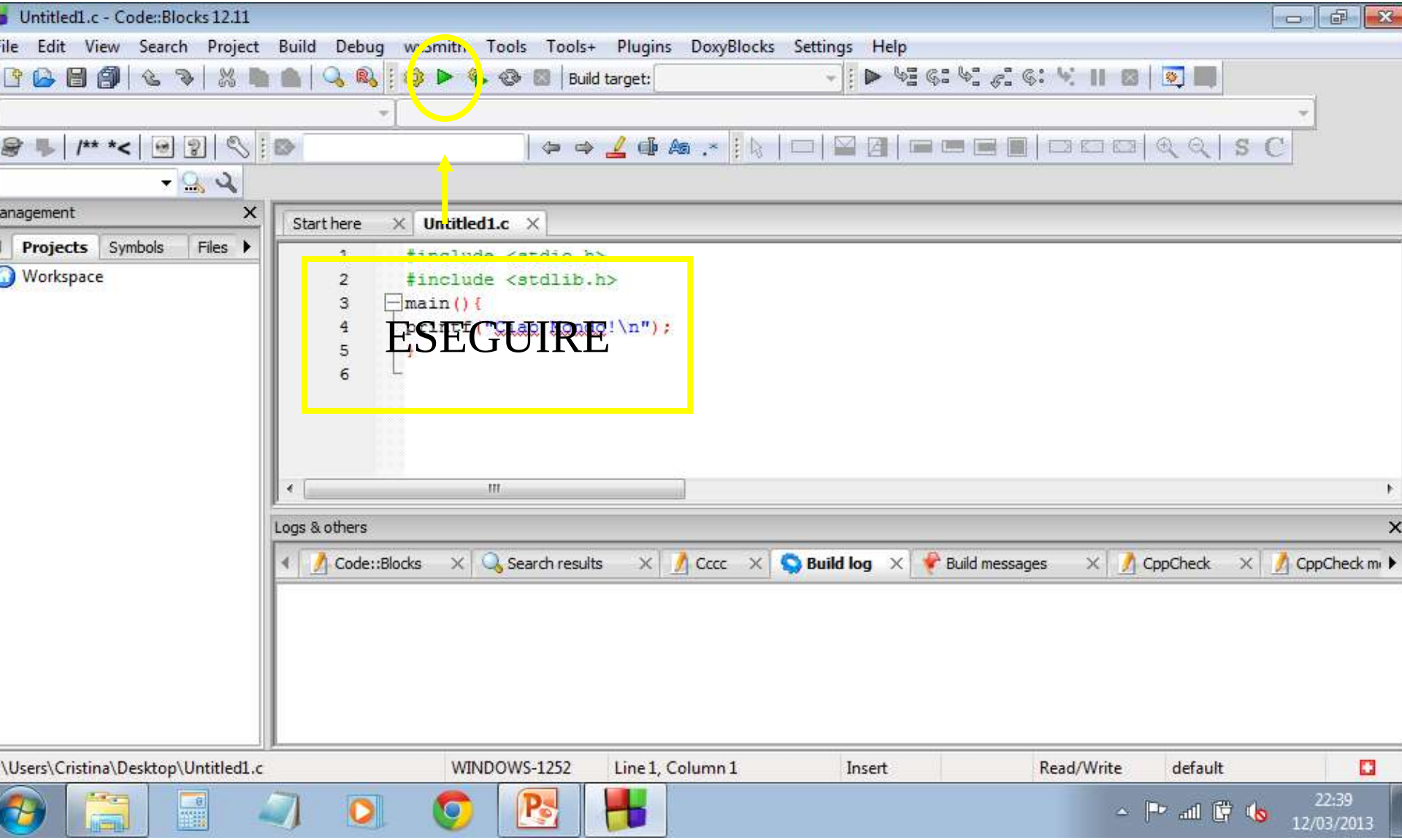
```
#include <iostream>

using namespace std;

main()
{ cout << "Ciao Mondo!";
}
```

Il file deve essere salvato con estensione .cpp


Untitled1.c - Code::Blocks 12.11
File Edit View Search Project Build Debug wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Build target:
Management
Projects Symbols Files
Workspace
Start here
Untitled1.c
#include <stdlib.h>
main(){
COMPILARE
Logs & others
Code::Blocks
Search results
Cccc
Build log
Build messages
CppCheck
\Users\Cristina\Desktop\Untitled1.c
WINDOWS-1252
Line 1, Column 1
Insert
Read/Write
default
22:39
12/03/2013


Untitled1.c - Code::Blocks 12.11
File Edit View Search Project Build Debug wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Build target:
Start here
Untitled1.c
#include <stdlib.h>
main(){
ESEGUIRE
Logs & others
Code::Blocks
Search results
Cccc
Build log
Build messages
CppCheck
\Users\Cristina\Desktop\Untitled1.c
WINDOWS-1252
Line 1, Column 1
Insert
Read/Write
default
22:39
12/03/2013

Opzioni di formattazione

Sequenza – Parola chiave	AZIONE
endl o \n	Va a capo
\t	Sposta a destra di un tab
\\	**Scrive **
\”	Scrive ”

Esercizi

- Scrivere 3 volte il proprio nome, andando a capo ogni volta
- Come sopra ma sfalsando la scrittura di uno spazio ogni riga
- Come sopra ma sfalsando la scrittura di una tabulazione
- Scrivere la frase Mario disse:‘‘Ciao a tutti!’’

Inserire un commento

Un commento è una frase che il compilatore non interpreterà come un'istruzione ma fornisce informazioni utili a chi leggerà o dovrà modificare il programma che abbiamo creato.

//COMMENTO oppure /* COMMENTO */

/* COMMENTO

SU

PIU' RIGHE */

Vocabolario del C++

- Lettere dell'alfabeto inglese (Case sensitive)
- Numeri (cifre decimali)
- Lo spazio bianco
- Simboli speciali:
  - operatori aritmetici + - * /

    % (modulo, rende il resto della divisione tra due interi)
  - operatori logici > < = >= <= && || !
  - caratteri speciali & ? !

Variabili

Una **variabile** è una locazione di memoria dove può essere immagazzinato un dato affinché possa essere usato durante l'elaborazione.

Le variabili devono essere **dichiarate** prima del loro utilizzo. E' necessario predisporre lo spazio di memoria prima di poterci registrare un dato.

Identificatori

Un **identificatore** è il nome simbolico che il programmatore assegna alle variabili.

- Sono costituiti da 1 o più caratteri di cui **il primo alfabetico** (meglio lettera minuscola) o _
- Case sensitive
- Lunghezza arbitraria (meglio meno di 31 caratteri)
- Nome evocativo
- Diversi dalle parole chiave (che sono sempre minuscole)
- Non possono contenere spazi

Dichiarazione delle variabili

E’ necessario scegliere l’identificatore e precisare il **tipo di dato** che la variabile dovrà contenere.

TIPI DI DATO PRIMARI

TIPO	**Descrizione**	**Memoria (byte)**	**Max**	**Min**
int	Numero intero	2	32767	-32768
float	Numero reale	4	$\pm 3{,}4 * 10^{38}$	$\pm 3{,}4 * 10^{-38}$
double	Numero reale	8	$\pm 1{,}7 * 10^{308}$	$\pm 1{,}7 * 10^{-308}$
char	carattere	1	255	0

Attenzione: A seconda della versione del compilatore usata il dato di tipo int può occupare 4 o 8 byte, avendo ,quindi, una variabilità maggiore di quella indicata.

Dichiarazione delle variabili

Il linguaggio C++ offre altri due tipi di variabile:

TIPO	Descrizione	Memoria (byte)	Valori possibili
bool	Variabile booleana	1	True-False
string	Stringa	Dipende dal contenuto	-

Per manipolare variabili di tipo stringa devo includere anche la libreria string
#include <string>

Attenzione: A seconda della versione del compilatore usata il dato di tipo bool può occupare 2 o più byte.

Dichiarazione delle variabili

Tipo di dato **Identificatore**

```
int a;
char x;
float n, m;
```

Assegnazione dei valori alle variabili

```
int a,b;
char c;
```

- Assegnazione diretta

```
int a=5;
a=5;
c='Z'
```

5

- Copia da un'altra variabile

```
b=a;
```

5

- Successivo ad un calcolo

```
b=2*a;
```

10

- Acquisito da tastiera

Istruzioni di input in C++

```
int a;
float b;

cin >> a;

cin >> b;
```

OPPURE

```
cin >> a >>b;
```

Suggerimento

Per testare la correttezza e la generalità di un programma è necessario testarlo, cioè mandarlo in esecuzione diverse volte con opportuni valori dell'input.

Può accadere che i dati acquisiti in una esecuzione rimangano memorizzati nell'esecuzione successiva è quindi opportuno ripulire l'INPUT attraverso la funzione

fflush(stdin);

da collocare prima della } di chiusura del main.

Il casting delle variabili

Le operazioni aritmetiche su variabili del medesimo tipo restituiscono un risultato del medesimo tipo.

Esempio:

```
int a=1, b=3;
float c;
c=a/b;
```

Mi aspetto che il contenuto di c sia 0,33333 ed invece è 0,00000!

Il casting delle variabili

Per risolvere il problema posso dichiarare anche a e b come float, oppure eseguire il casting della variabile.

Il **casting** di una variabile è una operazione che consiste nel cambiare temporaneamente (solo per quell'operazione) il tipo della variabile.

Il casting si effettua facendo precedere l'identificatore della variabile dal tipo racchiuso tra parentesi.

Il casting delle variabili

Esempio:

```
float a=1, b=3,c;
 c = a/b;
```

oppure

```
int a=1, b=3;
float c;
c = (float) a/b;
```

In entrambe i casi il contenuto di c è 0,33333!

Istruzioni di output in C++

Consentono di visualizzare sul monitor (nella finestra di comando) il valore di una variabile e/o scritte e simboli a scelta del programmatore.

```
cout << ‘‘frase o simboli a scelta’’;

cout << identificatore;

cout << ‘‘frase’’ << identificatore << ‘‘frase2’’;
```

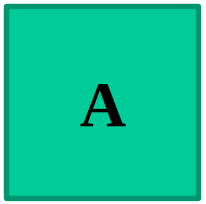
Istruzioni di I/O per i caratteri

char a=‘A’;

Attenzione: Ricordarsi di racchiudere tra apici singoli il carattere da assegnare alla variabile.

cout << a;

cin >> a;



Le costanti

Sono quantità non modificabili durante l'elaborazione.

Possono essere dichiarate in due modi diversi:

- Mediante una direttiva al preprocessore, da collocare dopo la chiamata alle librerie.

```
#include <iostream>

#define PIGRECO 3.14
```

- Mediante una variabile bloccata.

```
const float  PIGRECO=3.14;
```

Le costanti

- Mediante la direttiva al preprocessore.
  - Non alloco spazio in memoria
  - Coinvolgo il preprocessore i cui errori sono difficili da rilevare
- Mediante una variabile bloccata.
  - Alloco spazio in memoria.
  - E’ necessario quando si passa un parametro che non si vuole venga accidentalmente modificato.

Le costanti

L’identificatore PIGRECO può essere usato come gli identificatori delle variabili **ma non può essere riassegnato.**

a=PIGRECO;

PIGRECO =5;