

Esercitazione 1



UNIVERSITÀ
DEGLI STUDI
DI GENOVA

Dibris

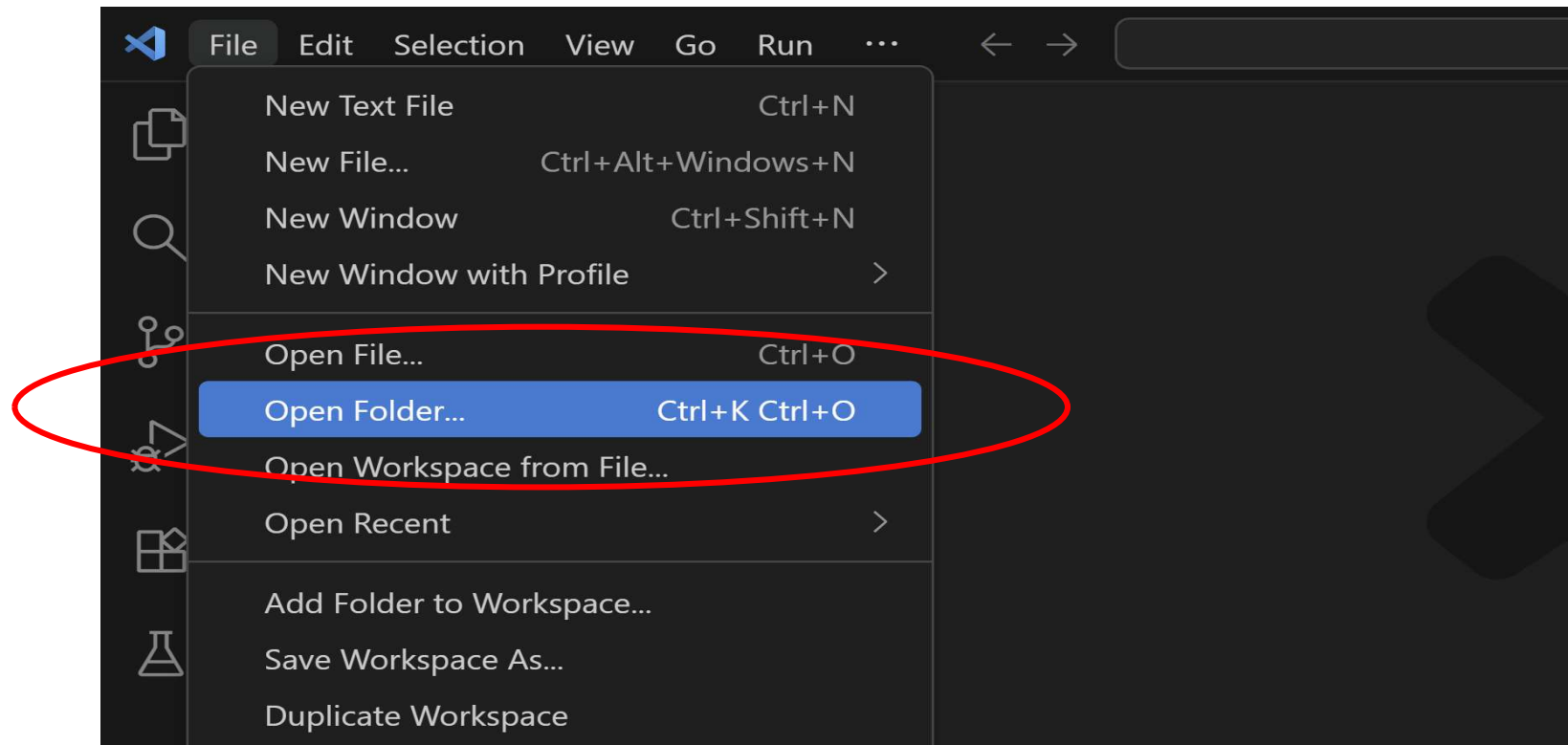
Gualtiero Volpe
gualtiero.volpe@unige.it

Esercizio 1.1

Sviluppare un programma in linguaggio C++ che chieda all'utente di immettere da tastiera la lunghezza di due cateti di un triangolo rettangolo (due numeri interi) e stampi a video il valore del quadrato dell'ipotenusa.

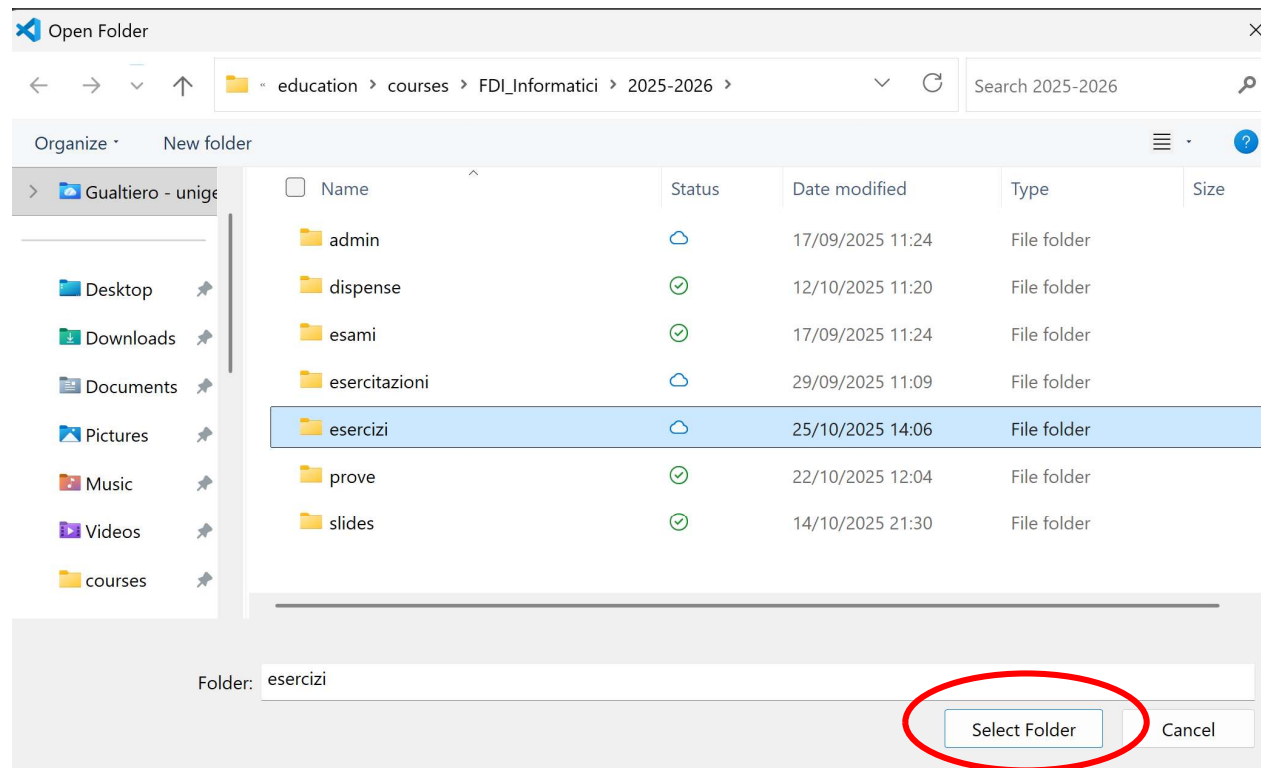
Scrivere, compilare ed eseguire un programma

Selezionare il menù **File** e fare clic su **Open Folder...**



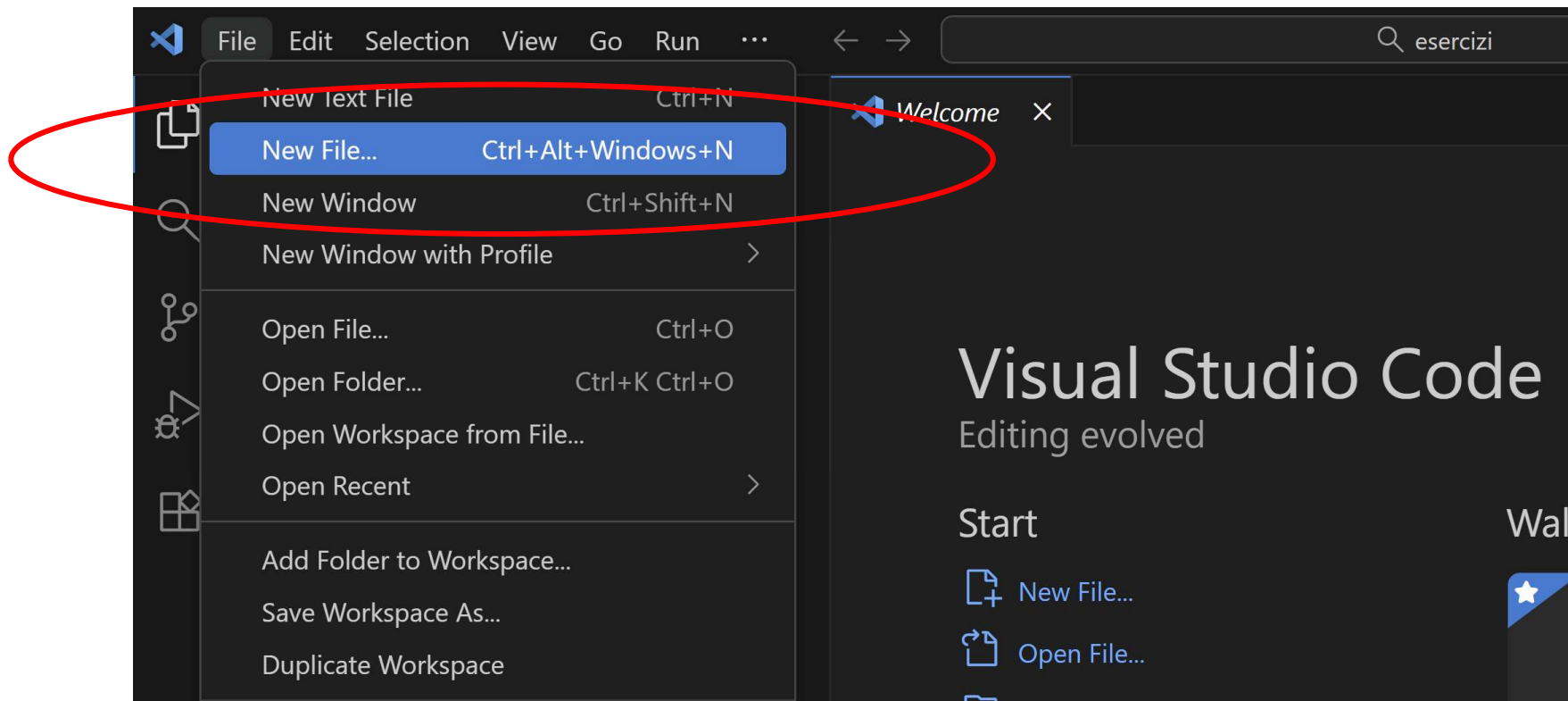
Scrivere, compilare ed eseguire un programma

Selezionare la cartella dove salvare il programma e fare clic su **Select Folder**.



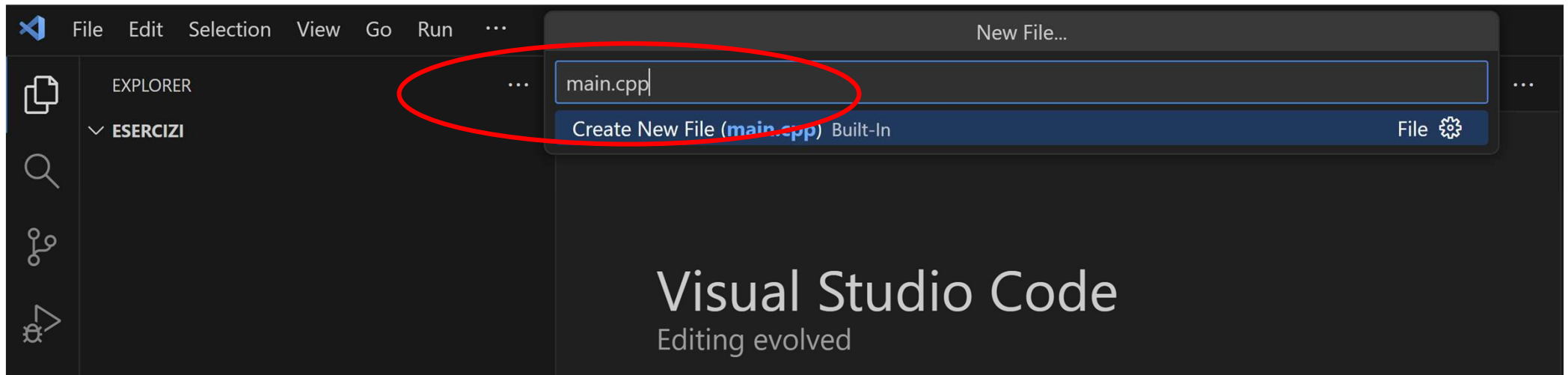
Scrivere, compilare ed eseguire un programma

Selezionare il menù **File** e fare clic su **New File...**



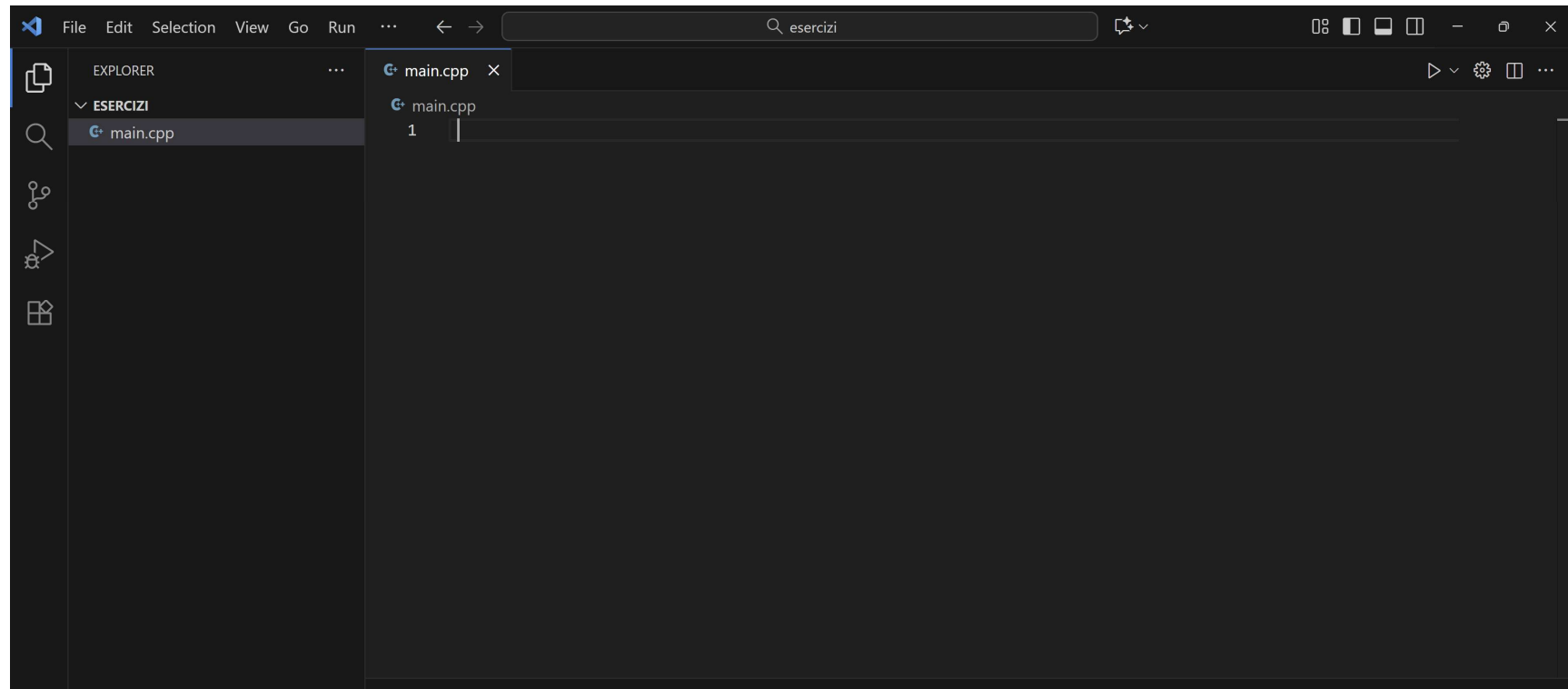
Scrivere, compilare ed eseguire un programma

Dare un nome al nuovo file che conterrà il codice sorgente del programma (ad esempio, *main.cpp*) e premere Invio.



Scrivere, compilare ed eseguire un programma

A questo punto, possiamo iniziare a sviluppare il programma!



Esercizio 1.1: Procedura per risolvere l'esercizio

1. Dichiarare due variabili intere per contenere la lunghezza dei cateti.
 2. Utilizzare gli oggetti di I/O cin e cout per ottenere i due numeri tramite il terminale, salvando il contenuto nelle variabili corrispondenti.
 3. Effettuare l'operazione di calcolo del quadrato dell'ipotenusa.
 4. Stampare il risultato sul terminale.
-

Esercizio 1.1: Soluzione

```
G++ main.cpp > main()
1  #include <iostream>
2
3  int main()
4  {
5      int cateto1 = 0, cateto2 = 0, quadrato_ipotenusa = 0;
6      std::cout << "Inserisci la lunghezza di un cateto: ";
7      std::cin >> cateto1;
8      std::cout << "Inserisci la lunghezza del secondo cateto: ";
9      std::cin >> cateto2;
10
11     quadrato_ipotenusa = cateto1 * cateto1 + cateto2 * cateto2;
12
13     std::cout << "Il quadrato dell'ipotenusa vale " << quadrato_ipotenusa << std::endl;
14     return 0;
15 }
```

Esercizio 1.1: Soluzione alternativa

```
G+ main.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int cateto1 = 0, cateto2 = 0, quadrato_ipotenusa = 0;
6      cout << "Inserisci la lunghezza di un cateto: ";
7      cin >> cateto1;
8      cout << "Inserisci la lunghezza del secondo cateto: ";
9      cin >> cateto2;
10
11     quadrato_ipotenusa = cateto1 * cateto1 + cateto2 * cateto2;
12
13     cout << "Il quadrato dell'ipotenusa vale " << quadrato_ipotenusa << endl;
14     return 0;
15 }
```

Esercizio 1.1: alcune note (già viste a lezione)

#include <iostream>

`#include` è una direttiva per il preprocessore, usata per includere file nel nostro programma (in particolare file che contengono librerie di codice).

In questo esempio, serve ad includere il contenuto del file *iostream*.

In particolare, questo ci serve per usare le librerie standard di I/O di C++ che mettono a disposizione del programmatore quanto necessario per la gestione delle operazioni di I/O.

Questo sistema si basa sul concetto di canale (stream) inteso come mezzo attraverso cui confluiscono le informazioni provenienti o inviate dai diversi dispositivi hardware (le unità esterne).

I canali standard di I/O sono “collegati” rispettivamente agli oggetti di I/O: **cout**, **cerr** e **cin** che permettono per mezzo degli operatori **<< (inserimento)** [i cui operandi di sinistra sono rispettivamente `cout` oppure `cerr`] e **>> (estrazione)** [il cui operando di sinistra è `cin`] di effettuare le operazioni di I/O.

Esercizio 1.1: alcune note (già viste a lezione)

using namespace std

In C++ esistono i namespace, ovvero costrutti che permettono di definire variabili e funzioni che siano globali, quindi visibili a tutte le funzioni, senza però entrare in conflitto con altre librerie che impiegano lo stesso nome di variabile o funzione.

Il **namespace std** definisce le variabili globali cin, cerr e cout.

Se quindi voglio utilizzarle, ho due possibilità:

1. Includere il file in cui sono definiti (*#include <iostream>*) e usare il namespace std (*using namespace std*) nel codice, così da poter direttamente utilizzare quelle variabili.
2. Includere il file in cui sono definiti (*#include <iostream>*) e fare riferimento specifico al loro namespace ogni volta che le utilizzo, usando l'operatore di risoluzione :: (es. **std::cout**)

La prima opzione **non** è considerata una best practice, ma può essere utilizzata soprattutto nelle prime esercitazioni per semplificare la scrittura del codice

Esercizio 1.1: alcune note (già viste a lezione)

```
int main() {...  
  
return 0;  
  
}
```

Ogni programma in C++ deve avere la funzione `main()`. Le parentesi graffe indicano l'inizio e la fine della funzione (e del programma). L'esecuzione del codice inizia da questa funzione.

A norma di standard, `main` deve avere tipo *int* e, se nel corpo della funzione non viene inserita esplicitamente un'istruzione `return`, il compilatore inserisce automaticamente `return 0`.

Esercizio 1.1: alcune note (già viste a lezione)

```
cout << "Il quadrato dell'ipotenusa vale " << quadrato_ipotenusa << endl;
```

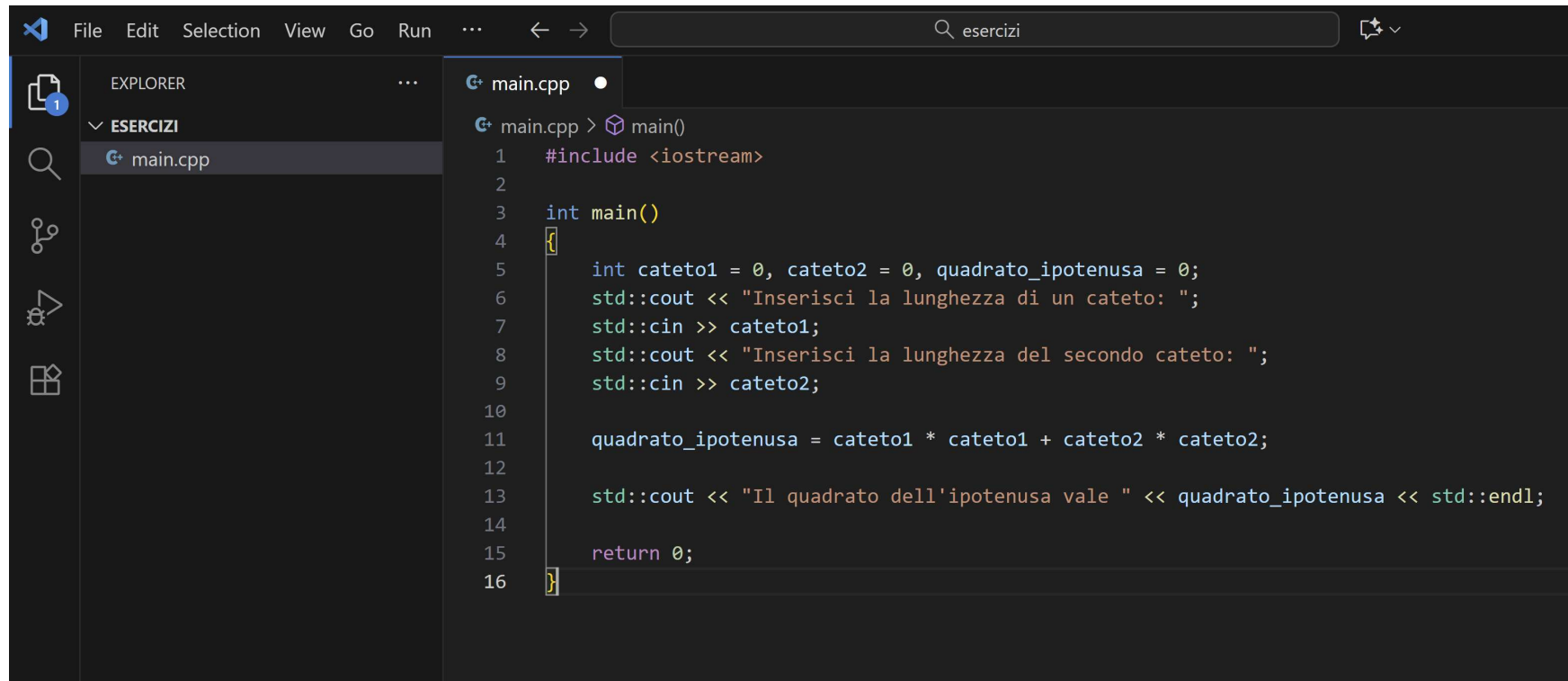
cout stampa il contenuto all'interno degli apici. Deve essere seguito dall'operatore << (inserimento), seguito dalla stringa che si vuole stampare a schermo. **endl** appartiene sempre al namespace std, serve per andare a capo, oltre che per assicurarsi che il buffer in uscita venga *scaricato*.

Due ulteriori aspetti importanti:

- In C++ la separazione tra istruzioni è indicata dal punto e virgola (;) che termina ciascuna di esse. La suddivisione del programma in più righe serve a rendere il programma più leggibile alle persone che lo leggono, per il compilatore la cosa non fa alcuna differenza.
- Le istruzioni sono leggermente spostate sulla destra. Il codice si dice «indentato». Anche questa è solo una questione di leggibilità, ma non è importante ai fini della compilazione.

Scrivere, compilare ed eseguire un programma

Scrivere il codice del programma nella finestra nominata *main.cpp*.

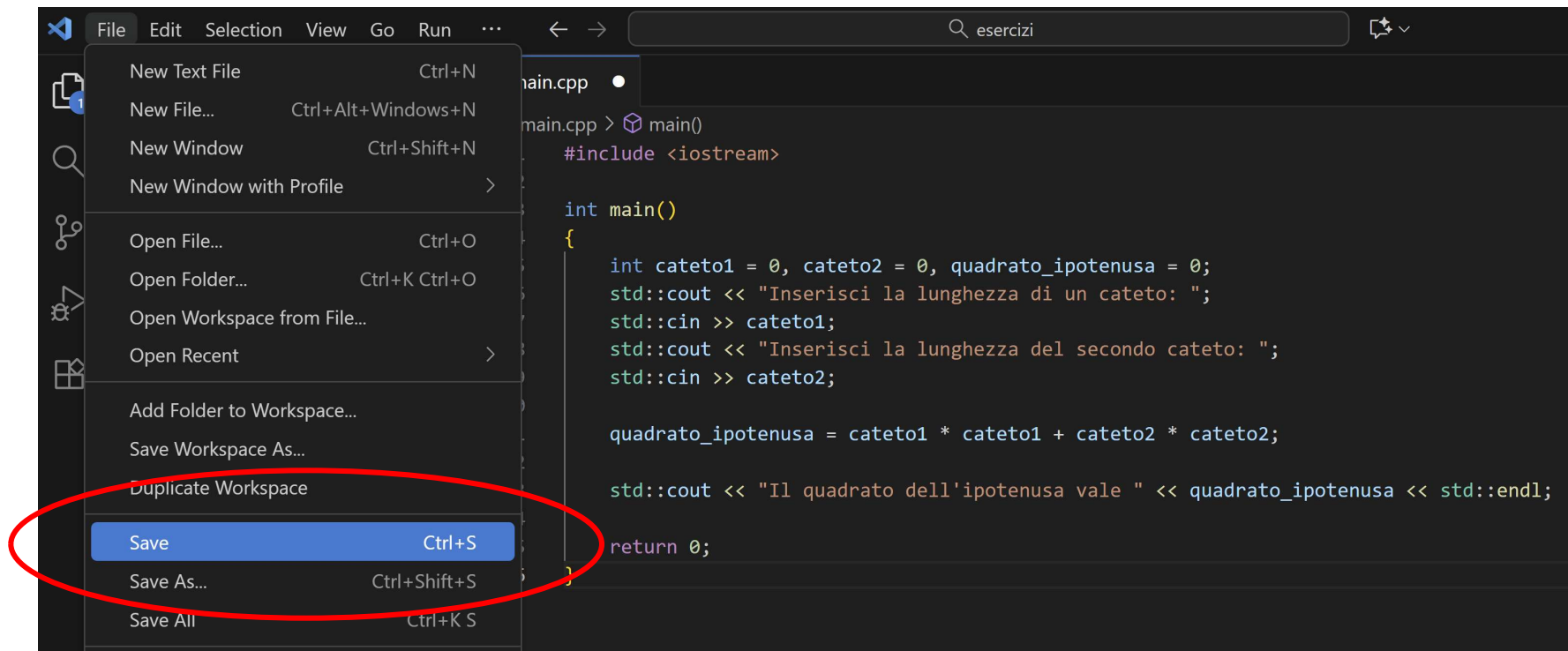


The screenshot shows the Visual Studio Code interface with a dark theme. The Explorer sidebar on the left shows a folder named 'ESERCIZI' containing a file 'main.cpp'. The main editor window displays the code for 'main.cpp'. The code is a C++ program that calculates the hypotenuse of a right triangle given two catheti. It includes the `<iostream>` header, defines the `main()` function, declares variables `cateto1`, `cateto2`, and `quadrato_ipotenusa`, prompts the user for input, calculates the hypotenuse using the Pythagorean theorem, and prints the result.

```
1  #include <iostream>
2
3  int main()
4  {
5      int cateto1 = 0, cateto2 = 0, quadrato_ipotenusa = 0;
6      std::cout << "Inserisci la lunghezza di un cateto: ";
7      std::cin >> cateto1;
8      std::cout << "Inserisci la lunghezza del secondo cateto: ";
9      std::cin >> cateto2;
10
11     quadrato_ipotenusa = cateto1 * cateto1 + cateto2 * cateto2;
12
13     std::cout << "Il quadrato dell'ipotenusa vale " << quadrato_ipotenusa << std::endl;
14
15     return 0;
16 }
```

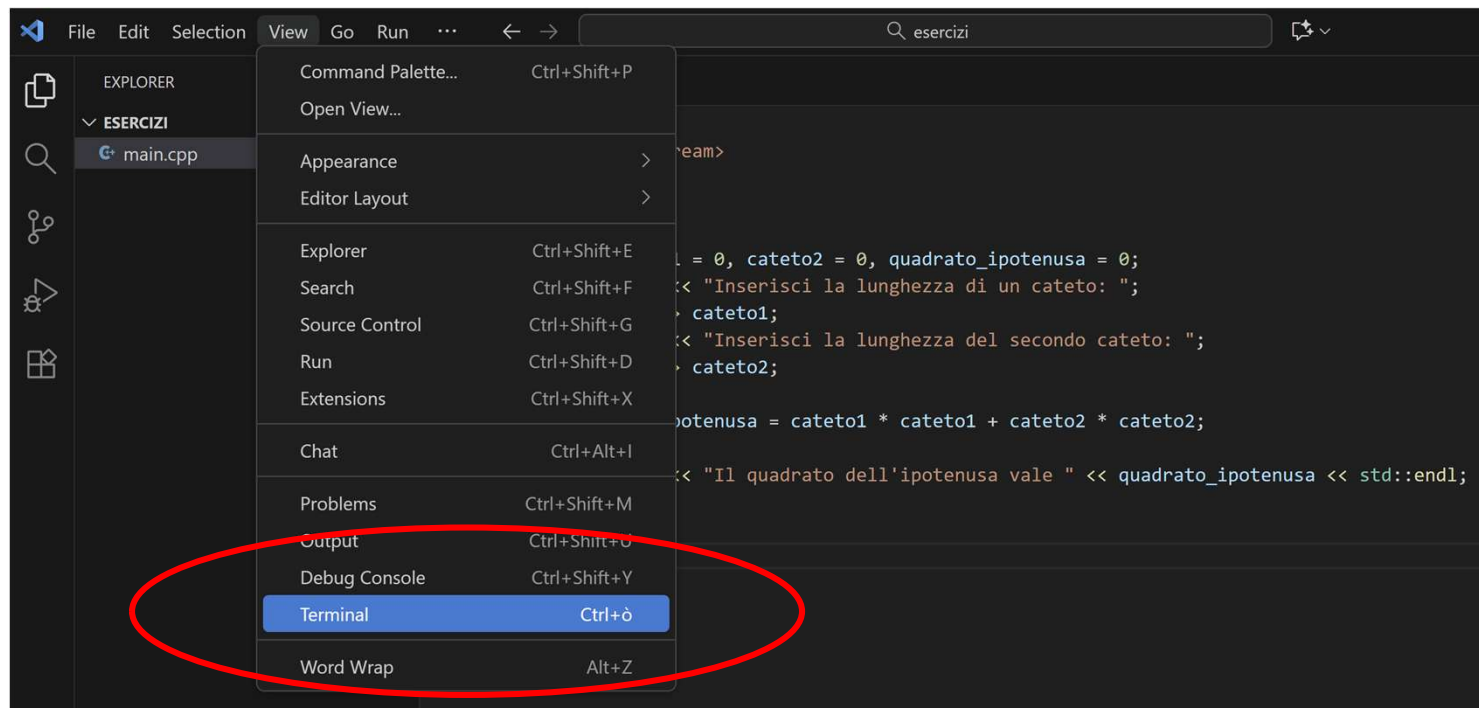
Scrivere, compilare ed eseguire un programma

Ricordarsi di salvare il lavoro, selezionando **File -> Save** o premendo **Ctrl+S**.



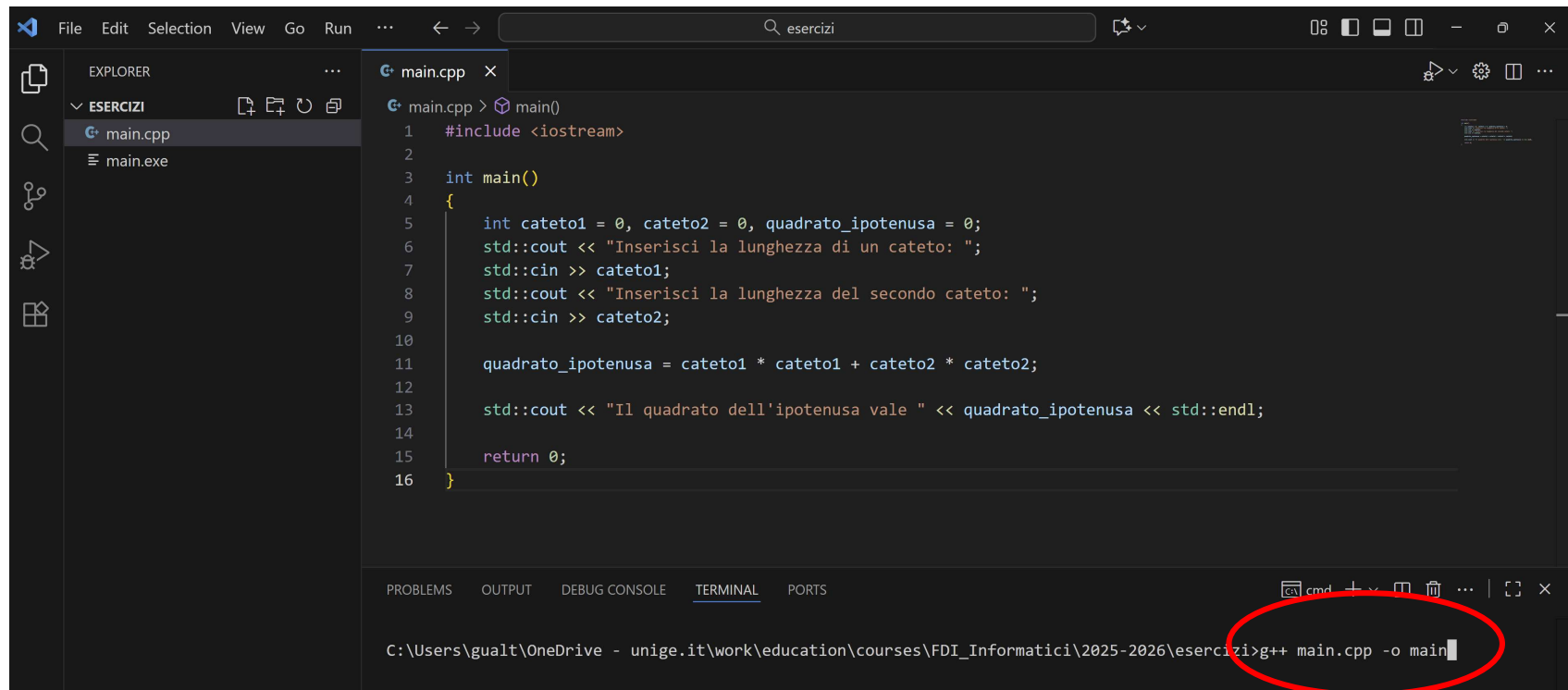
Scrivere, compilare ed eseguire un programma

Aprire la finestra del terminale integrato in Visual Studio Code, selezionando il menu **View** e facendo clic su **Terminal**.



Scrivere, compilare ed eseguire un programma

Nel terminale scrivere `g++ <file_sorgente> -o <file_eseguibile>` e premere Invio.
Nel caso del programma che stiamo sviluppando: `g++ main.cpp -o main`



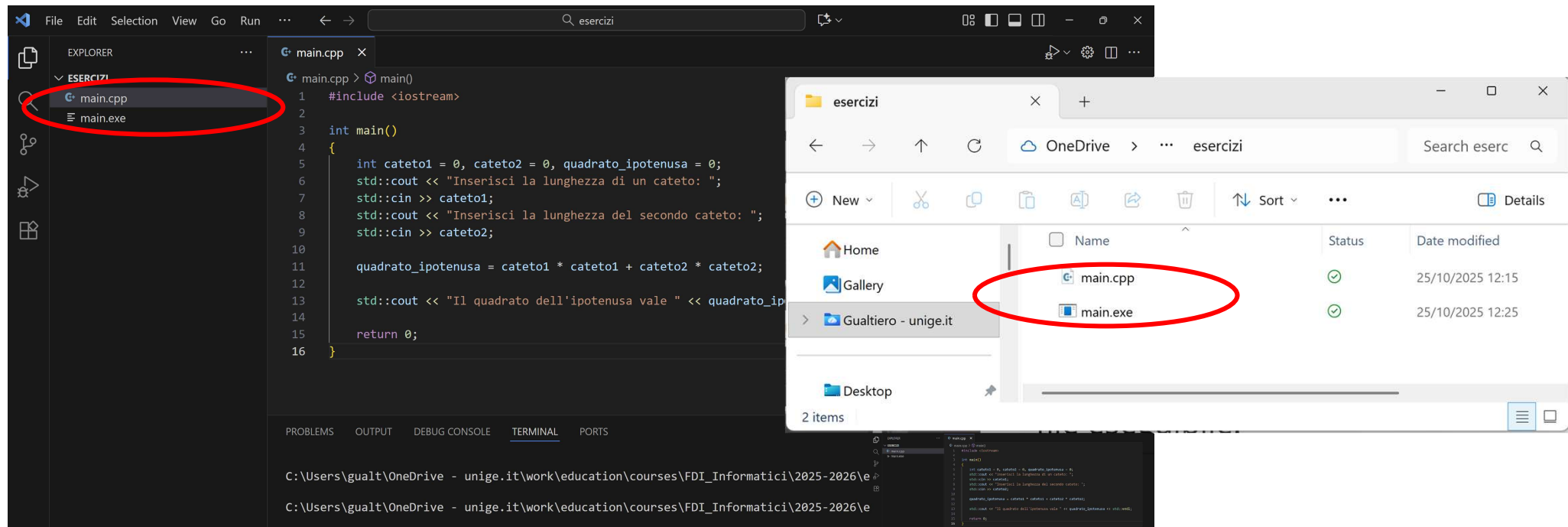
The screenshot shows the Visual Studio Code interface. On the left, the Explorer pane shows a project named 'ESERCIZI' containing 'main.cpp' and 'main.exe'. The main editor displays the content of 'main.cpp', which is a C++ program for calculating the hypotenuse. The code includes `<iostream>`, defines `main()`, and uses `std::cout` and `std::cin` for input and output. The calculation is `quadrato_ipotenusa = cateto1 * cateto1 + cateto2 * cateto2;`. The bottom panel shows the TERMINAL tab with the command `C:\Users\gualt\OneDrive - unige.it\work\education\courses\FDI_Informatici\2025-2026\esercizi>g++ main.cpp -o main` entered, which is circled in red.

```
1 #include <iostream>
2
3 int main()
4 {
5     int cateto1 = 0, cateto2 = 0, quadrato_ipotenusa = 0;
6     std::cout << "Inserisci la lunghezza di un cateto: ";
7     std::cin >> cateto1;
8     std::cout << "Inserisci la lunghezza del secondo cateto: ";
9     std::cin >> cateto2;
10
11     quadrato_ipotenusa = cateto1 * cateto1 + cateto2 * cateto2;
12
13     std::cout << "Il quadrato dell'ipotenusa vale " << quadrato_ipotenusa << std::endl;
14
15     return 0;
16 }
```

C:\Users\gualt\OneDrive - unige.it\work\education\courses\FDI_Informatici\2025-2026\esercizi>g++ main.cpp -o main

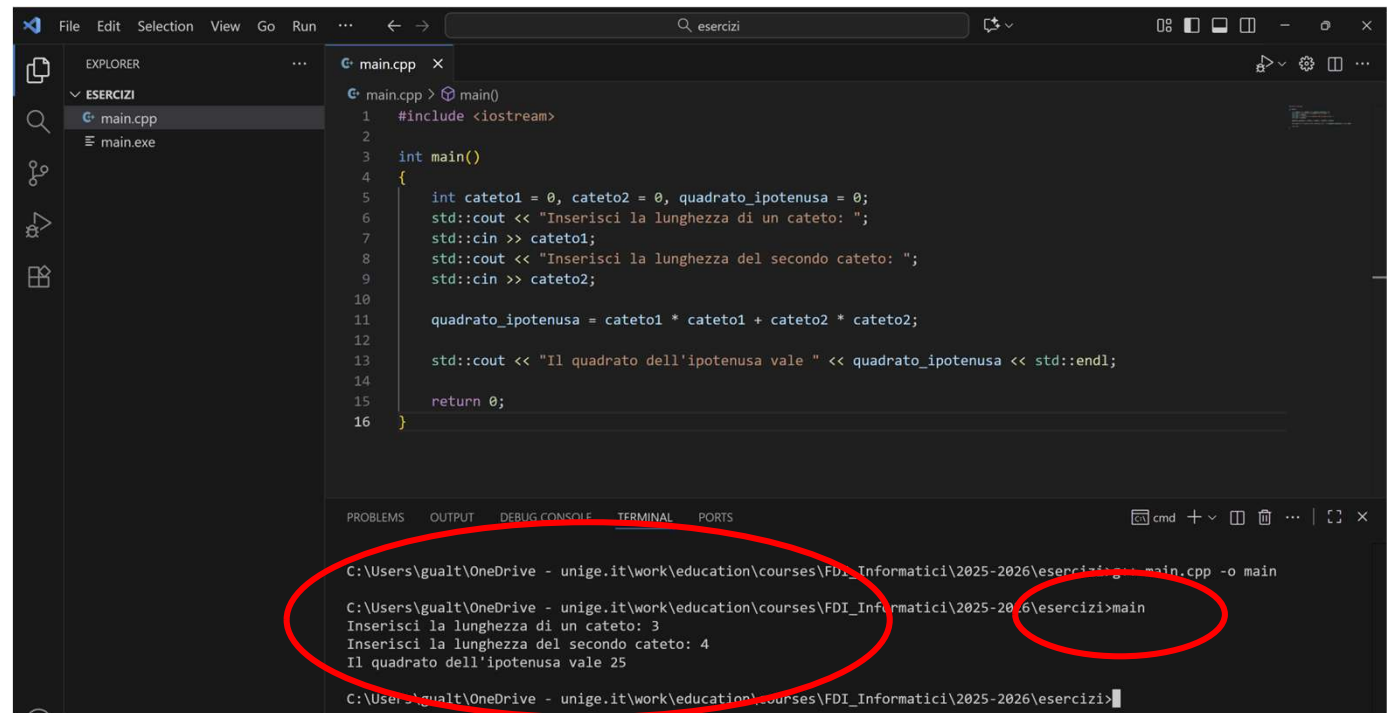
Scrivere, compilare ed eseguire un programma

Il compilatore compilerà il programma e, se non ci sono errori, genererà un file eseguibile nella stessa cartella che contiene il file sorgente.



Scrivere, compilare ed eseguire un programma

Per eseguire il programma, scrivere il nome del programma (in Windows) o `./<nome_del_programma>` (in Linux o MacOS) nel terminale, nel nostro caso `main` oppure `./main`.

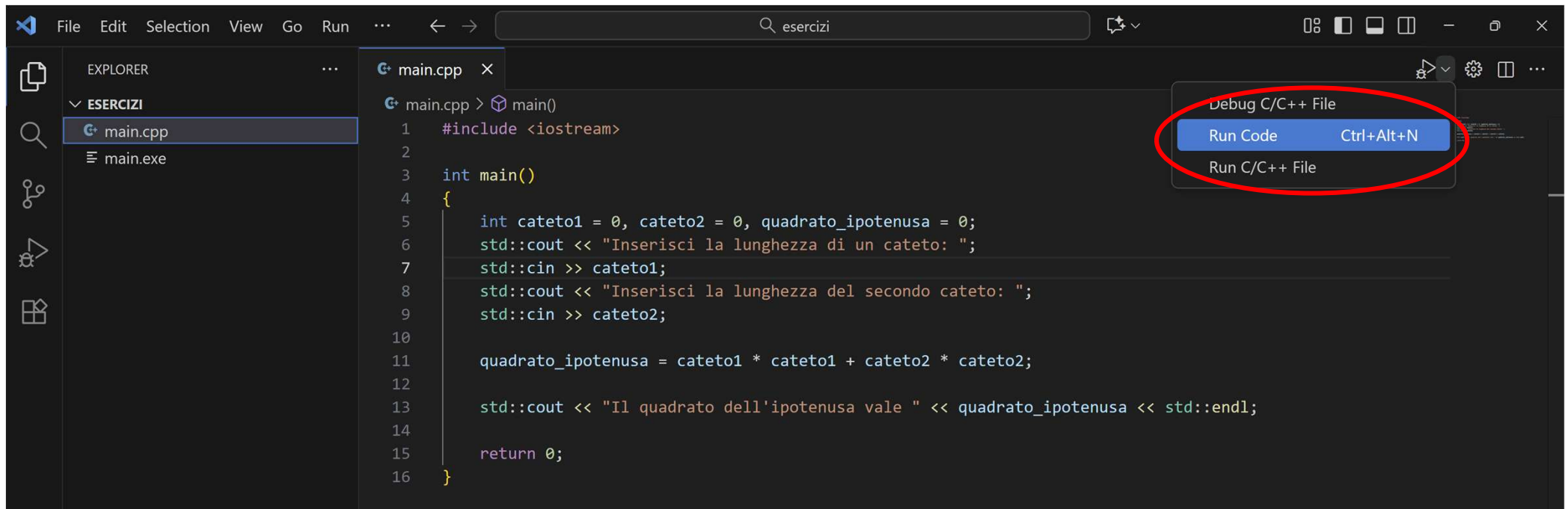


The screenshot shows the Visual Studio Code interface. The Explorer panel on the left shows a project named 'ESERCIZI' with files 'main.cpp' and 'main.exe'. The main editor displays the content of 'main.cpp', which is a C++ program that calculates the sum of squares of two integers. The terminal at the bottom shows the compilation and execution of the program. The compilation command is circled in red: `C:\Users\gualt\OneDrive - unige.it\work\education\courses\FDI_Informatici\2025-2026\esercizi> g++ main.cpp -o main`. The execution command is also circled in red: `C:\Users\gualt\OneDrive - unige.it\work\education\courses\FDI_Informatici\2025-2026\esercizi> main`. The program's output is shown below the execution command: `Inserisci la lunghezza di un cateto: 3`, `Inserisci la lunghezza del secondo cateto: 4`, and `Il quadrato dell'ipotenusa vale 25`.

```
File Edit Selection View Go Run ... < > esercizi
EXPLORER
  ESERCIZI
    main.cpp
    main.exe
main.cpp
1  #include <iostream>
2
3  int main()
4  {
5      int cateto1 = 0, cateto2 = 0, quadrato_ipotenusa = 0;
6      std::cout << "Inserisci la lunghezza di un cateto: ";
7      std::cin >> cateto1;
8      std::cout << "Inserisci la lunghezza del secondo cateto: ";
9      std::cin >> cateto2;
10
11     quadrato_ipotenusa = cateto1 * cateto1 + cateto2 * cateto2;
12
13     std::cout << "Il quadrato dell'ipotenusa vale " << quadrato_ipotenusa << std::endl;
14
15     return 0;
16 }
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
C:\Users\gualt\OneDrive - unige.it\work\education\courses\FDI_Informatici\2025-2026\esercizi> g++ main.cpp -o main
C:\Users\gualt\OneDrive - unige.it\work\education\courses\FDI_Informatici\2025-2026\esercizi> main
Inserisci la lunghezza di un cateto: 3
Inserisci la lunghezza del secondo cateto: 4
Il quadrato dell'ipotenusa vale 25
C:\Users\gualt\OneDrive - unige.it\work\education\courses\FDI_Informatici\2025-2026\esercizi>
```

Scrivere, compilare ed eseguire un programma

Opzionale: per compilare ed eseguire il programma, si può usare **CodeRunner**. Per farlo, selezionare il tasto freccia (▶) in alto a destra e fare clic su **Run Code**. Il programma sarà compilato ed eseguito nella finestra del terminale.



Esercizi da Svolgere

Esercizio 1.2 – **Media di cinque numeri**

Sviluppare in linguaggio C++ un programma che chieda all'utente di immettere da tastiera cinque numeri interi e ne stampi a video la media (approssimandola ad un numero intero)

Esercizio 1.3 – **Divisione con resto**

Sviluppare in linguaggio C++ un programma che chieda all'utente di immettere due numeri interi, divida il primo numero per il secondo, e stampi a video il quoziente e il resto. Supporre, per semplicità, che il secondo numero non sia zero.

Esercizi da Svolgere

Esercizio 1.4 – **Convertitore di valuta**

Realizzare in linguaggio C++ un convertitore Euro -> Lire, utilizzando come fattore di conversione il numero intero costante 1936.

Esercizio 1.5 – **Esprimere una durata temporale in modi diversi**

Si scriva un programma in linguaggio C++ che chieda all'utente di immettere da tastiera un numero intero per esprimere una durata misurata in secondi e calcoli e stampi a video la stessa durata espressa in numero di settimane, di giorni, di ore, di minuti e di secondi.

Esercizi da Svolgere

Esercizio 1.6 – **Precedente e successivo**

Si scriva un programma in linguaggio C++ che chieda all'utente di immettere da tastiera un numero intero e stampi a video il numero immediatamente precedente e il numero immediatamente successivo. Si utilizzino gli operatori di incremento e decremento unitario.

Esercizio 1.7 – **Uguali o diversi?**

Si scriva un programma in linguaggio C++ che chieda all'utente di immettere da tastiera due numeri interi e stampi a video 0 se i due numeri sono diversi o un numero diverso da 0 se sono uguali. Si usino solo gli operatori di confronto.

Istruzioni Finali

Troverete le soluzioni di questi esercizi - con qualche commento aggiuntivo - qui:

https://github.com/gualtierovolpe/fondamenti_di_informatica_2025-26

Le soluzioni delle esercitazioni vengono rilasciate il giorno prima dell'esercitazione successiva.
