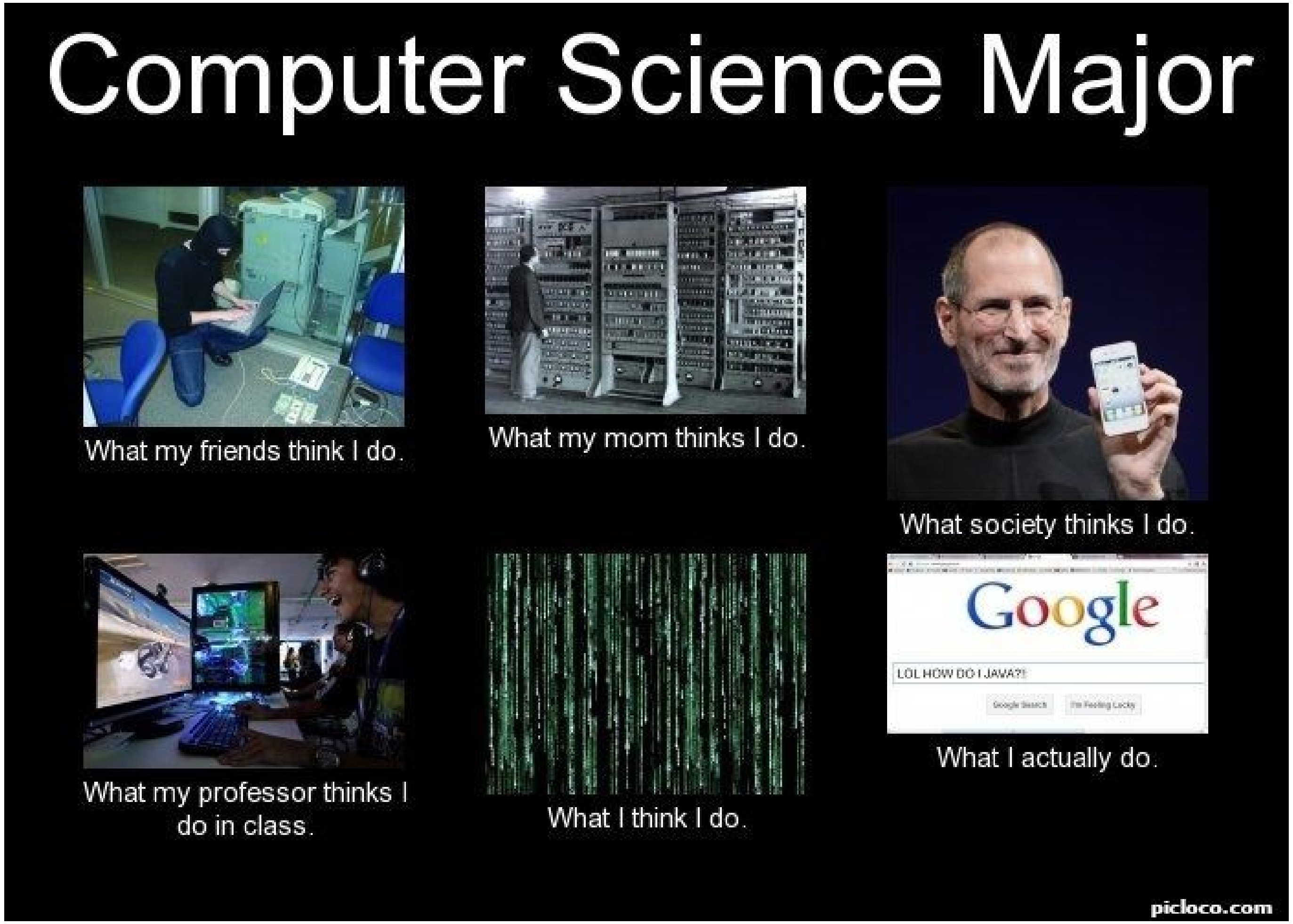

Computer Science Major
What my friends think I do.
What my mom thinks I do.
What society thinks I do.
What my professor thinks I do in class.
What I think I do.
Google
LOL HOW DO I JAVA?!
Google Search
I'm Feeling Lucky
What I actually do.
picloco.com

CS1428
Foundation of Computer Science

Bonus Content: Creating Libraries

Bonus Content

- This content will not be included on the final.
- This content will not be included in any lab or assignment.
- There is not “Check On Learning” for this module.

Why We Put Code into Libraries

- Libraries make general-purpose code easy include in multiple projects.
- Libraries are easy to transport.
- Libraries can be pre-compiled to obscure the source code from another programmer while exposing the interface.

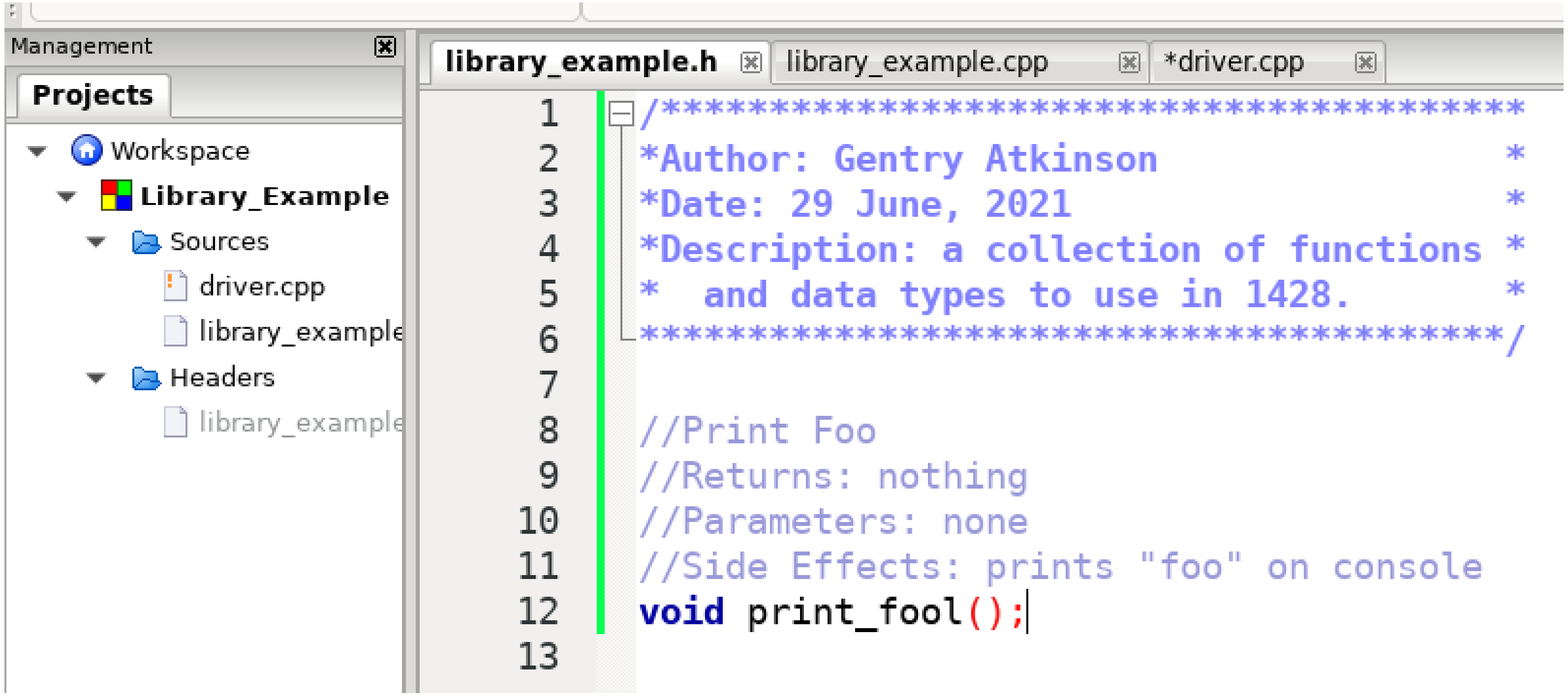
How We Put Code into Libraries

- We split our code into two files:
  - Header: saved with a .h extension. This file contains function prototypes and data type definitions. We should also include explanations for how to use every function and data type.
  - Source: saved with a .cpp extension. This file contains all of the definitions and implementations required to make the functions and data types in our header files work.
- The header file shows the “interface” of our code, i.e. how to use everything.
- The source file can be pre-compiled.

Libraries in Code::Blocks

- Local libraries need to be compiled and included along with your main file.
- Projects automatically link your local libraries at build time
- Local libraries are included using “library name.h” rather than <library name>

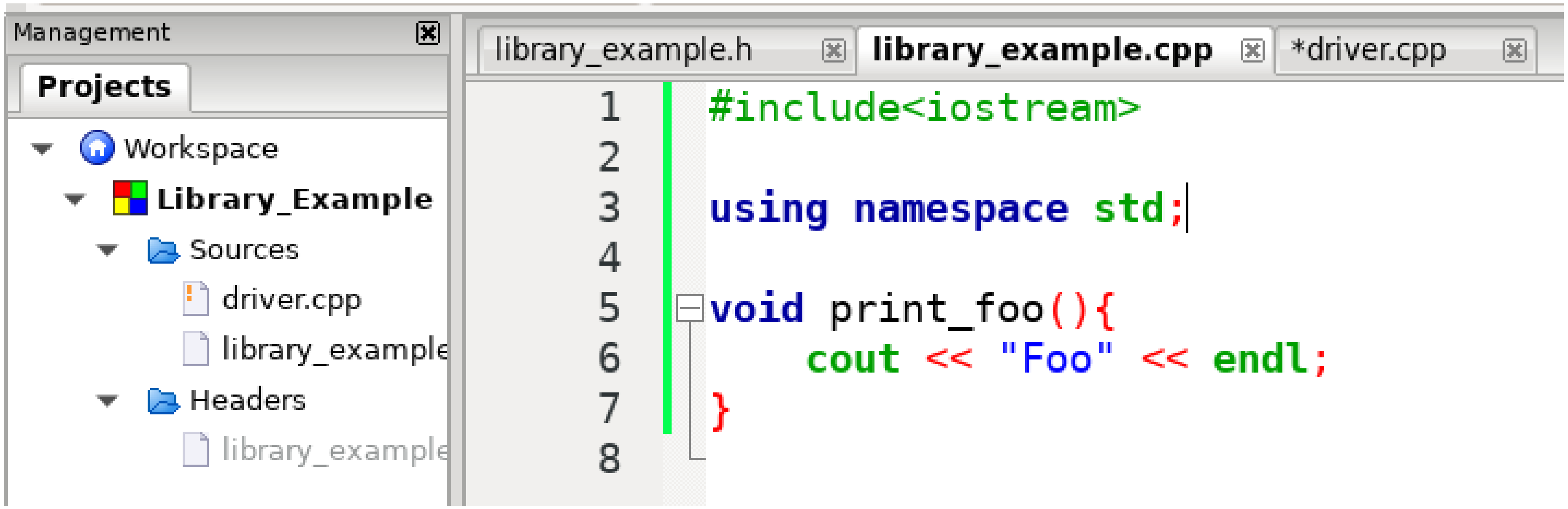
Writing Libraries



```
/****************************************
*Author: Gentry Atkinson                *
*Date: 29 June, 2021                    *
*Description: a collection of functions *
*  and data types to use in 1428.       *
****************************************/

//Print Foo
//Returns: nothing
//Parameters: none
//Side Effects: prints "foo" on console
void print_fool();
```

Writing Libraries



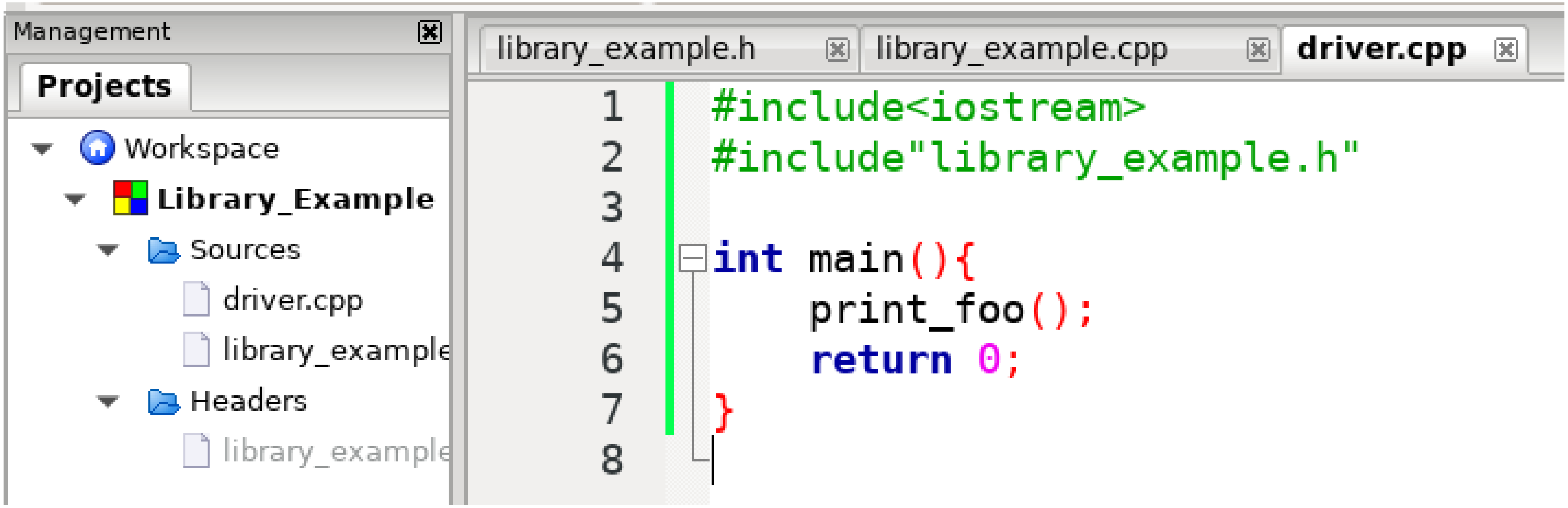
```
#include<iostream>

using namespace std;

void print_foo(){
    cout << "Foo" << endl;
}

```

Writing Libraries

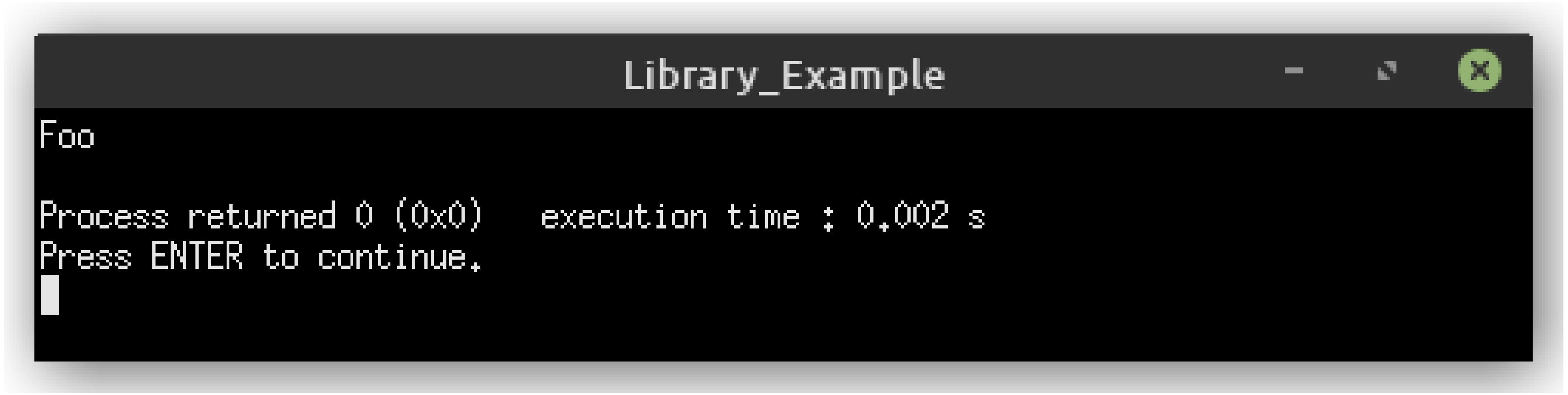


```cpp
#include<iostream>
#include"library_example.h"

int main(){
    print_foo();
    return 0;
}

```

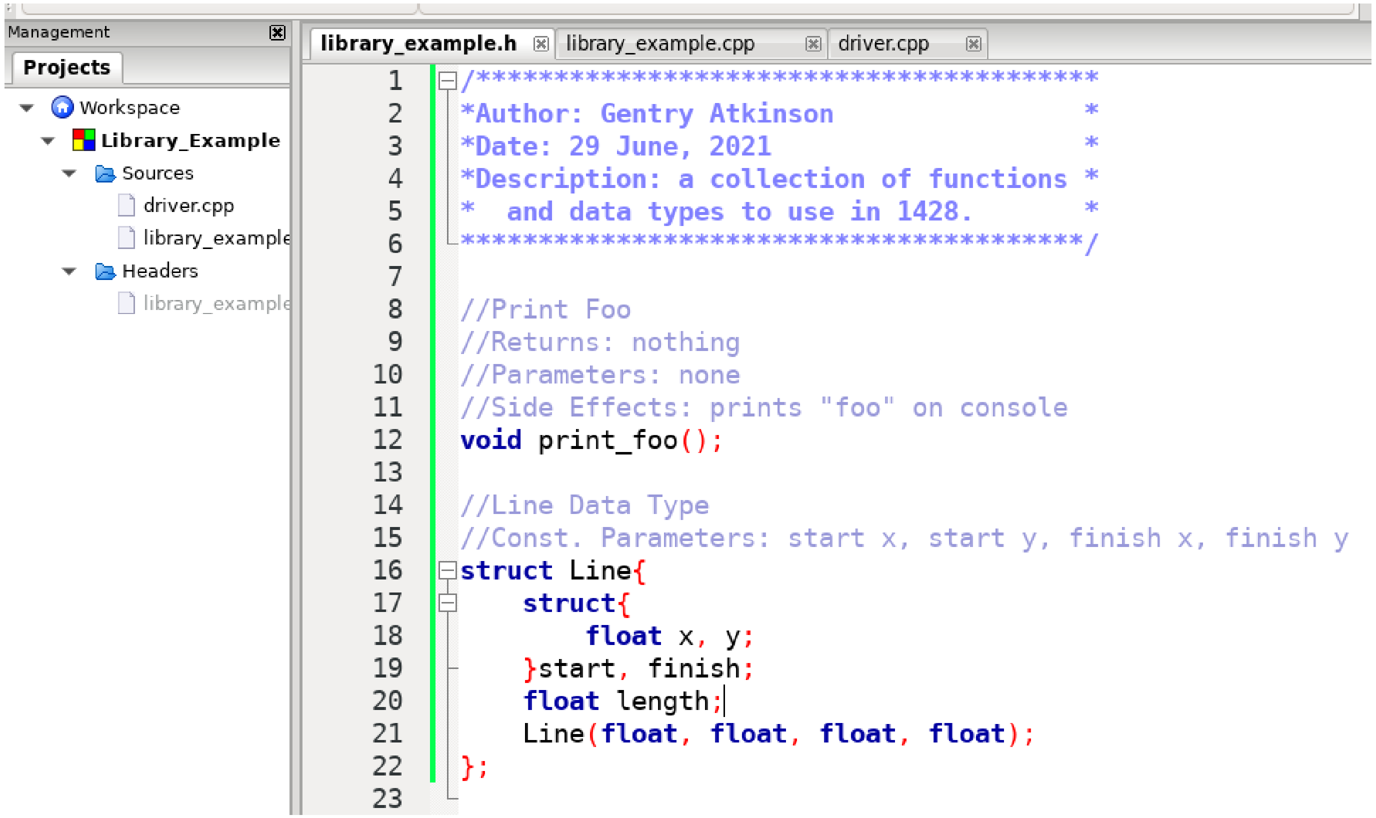
Writing Libraries



```
Foo

Process returned 0 (0x0)   execution time : 0.002 s
Press ENTER to continue.
```

Data Types in Libraries

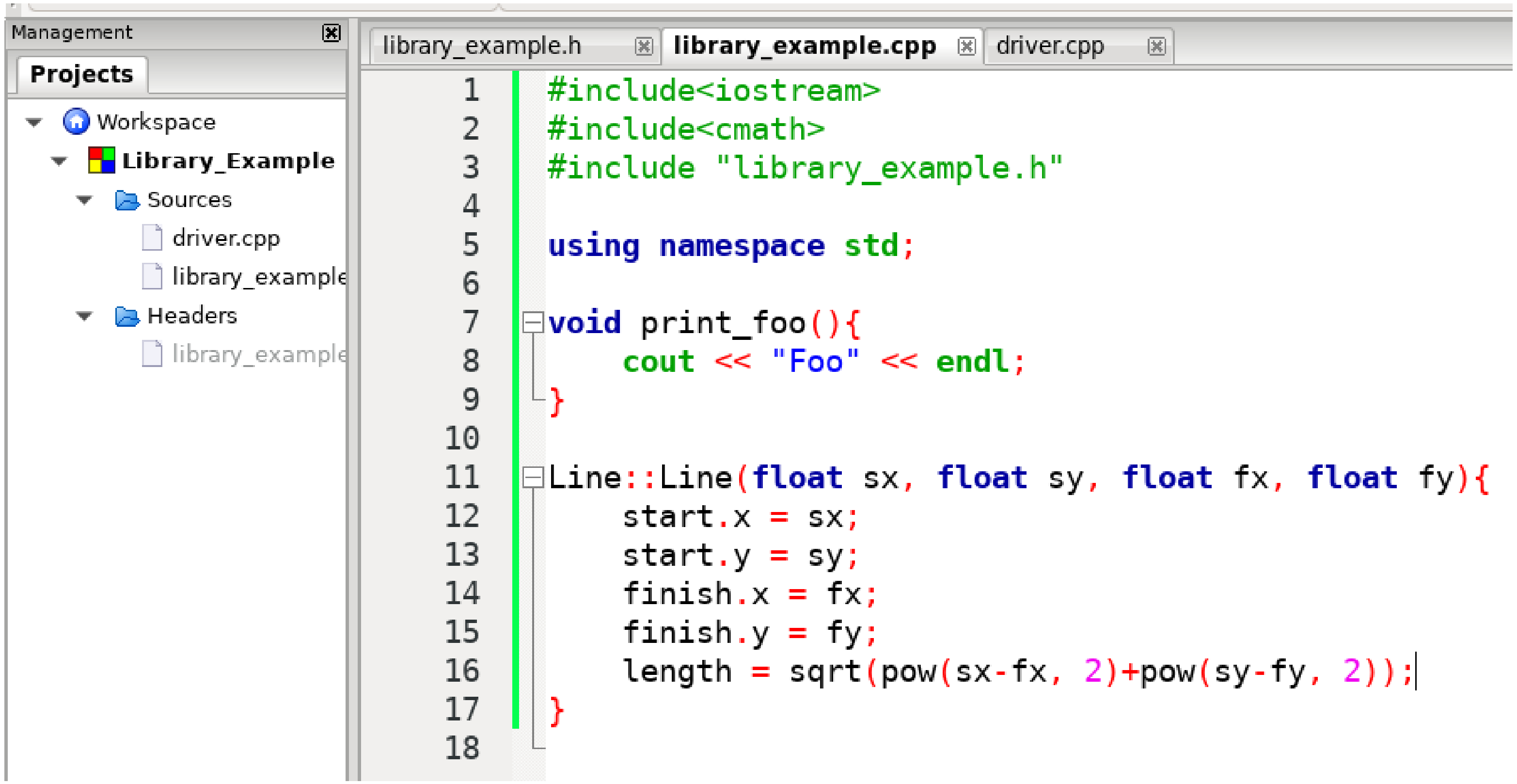


```cpp
/************************************
*Author: Gentry Atkinson             *
*Date: 29 June, 2021                 *
*Description: a collection of functions *
*   and data types to use in 1428.    *
************************************/

//Print Foo
//Returns: nothing
//Parameters: none
//Side Effects: prints "foo" on console
void print_foo();

//Line Data Type
//Const. Parameters: start x, start y, finish x, finish y
struct Line{
    struct{
        float x, y;
    }start, finish;
    float length;
    Line(float, float, float, float);
};
```

Data Types in Libraries



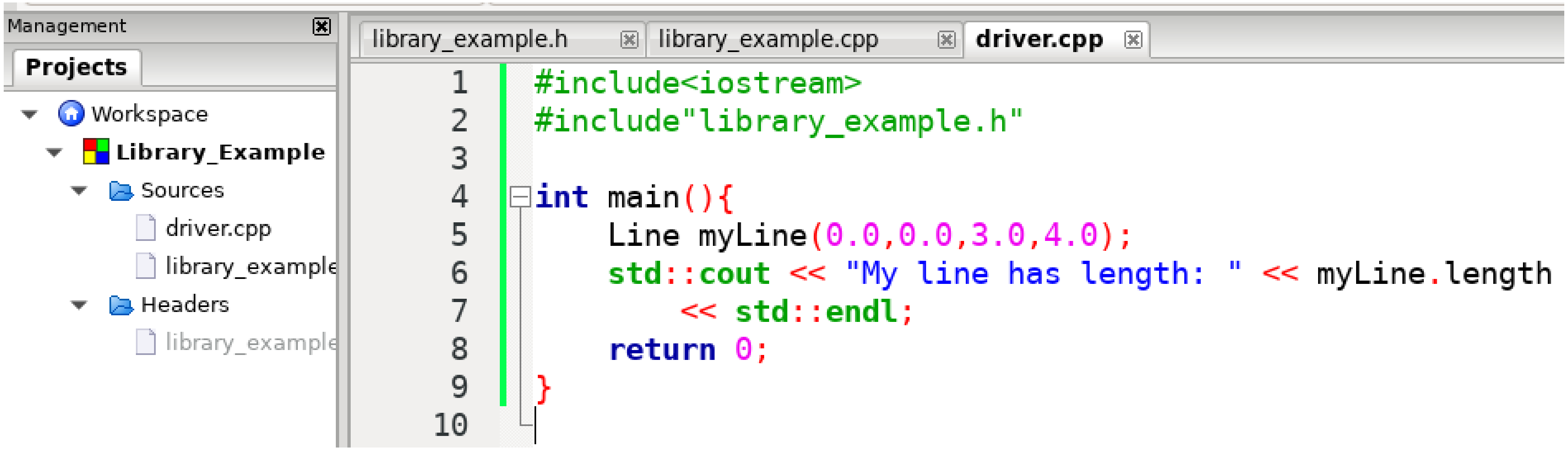
```cpp
#include<iostream>
#include<cmath>
#include "library_example.h"

using namespace std;

void print_foo(){
    cout << "Foo" << endl;
}

Line::Line(float sx, float sy, float fx, float fy){
    start.x = sx;
    start.y = sy;
    finish.x = fx;
    finish.y = fy;
    length = sqrt(pow(sx-fx, 2)+pow(sy-fy, 2));
}
```

Data Types in Libraries



Management

Projects

- Workspace
  - Library_Example
    - Sources
      - driver.cpp
      - library_example
    - Headers
      - library_example

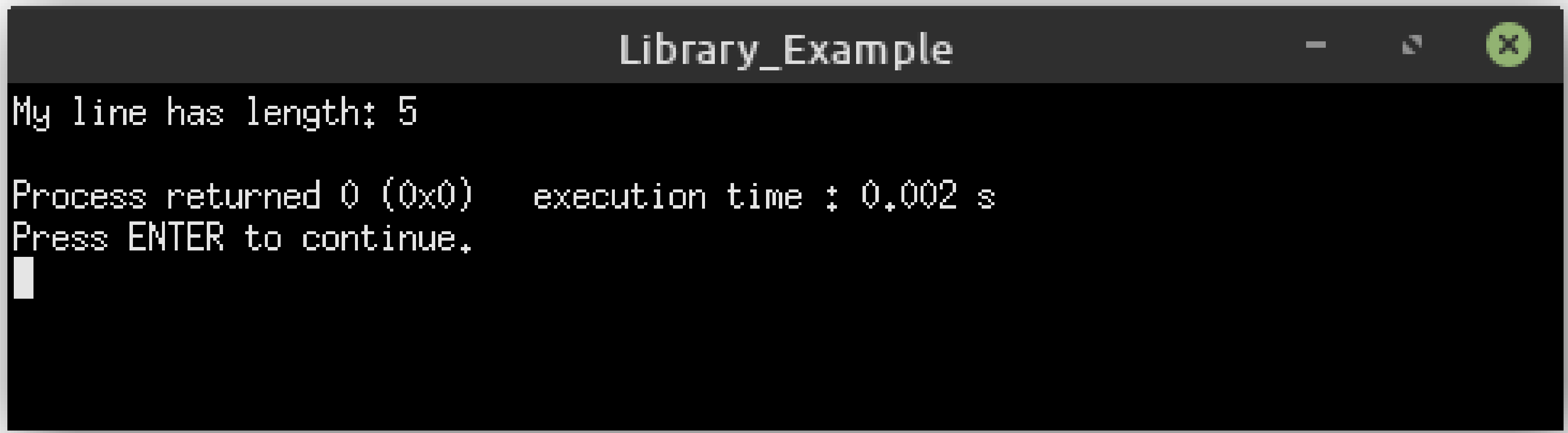
library_example.h | library_example.cpp | driver.cpp

```
#include<iostream>
#include"library_example.h"

int main(){
    Line myLine(0.0,0.0,3.0,4.0);
    std::cout << "My line has length: " << myLine.length
        << std::endl;
    return 0;
}

```

Data Types in Libraries



```
My line has length: 5

Process returned 0 (0x0)   execution time : 0.002 s
Press ENTER to continue.
```

When to Move Code to a Library

- The first code file that your reader reads should be relatively short.
- Header files represent long and complicated code as simple interfaces that are easy to understand and remember.
- Libraries let us logically encapsulate functions and data types in the same way that data types let us logically encapsulate variables.
- Code that is very likely to be re-used should be in a library.