
Jakob ☘ \u0000
@jcsrb
6 hours of debugging can save you 5 minutes of reading documentation

6.1: Multi File Development

CS2308
Gentry Atkinson

Why split up a program?

- Very large files can be hard to read.
- Putting many classes and function in one file can be confusing.
- We might not distribute source code to protect IP.

Including User Defined Libraries

- We can still use the **#include** pre-processor directive to use our own libraries.
- Only header (.h) files need to be included.
- Use the full filename to include a local library (e.g. “myClass.h”)

How are programs divided in OOP?

- Put each class definition in a separate *header* file.
  - The extension of a header is .h
- Put the implementation of each class in a separate .cpp file.
- Every program needs a main.cpp *driver.*

Example 1

In Employee.h

```
class Employee{
    private:
        string name;
        string title;
        float hourlyRate;
    public:
        Employee();
        Employee(const Employee&);
};
```

In Employee.cpp

```
Employee::Employee(){
    name = "";
    title = "";
    hourlyRate = 0;
}
Employee::Employee(const Employee& e){
    name = e.name;
    title = e.title;
    hourlyRate = e.hourlyRate;
}
```

Using local libraries

- Once a library is included in your program, you can use any functions or classes as if they were defined in the same file.
- You should have a “main.cpp” with a main function. This is called a driver.

Example 2

In main.cpp

```
#include<iostream>
#include “Employee.h”
using namespace std;

int main(int argc, char** argv){
    Employee a;
    Employee b(a);
    cout << "Opps. No getters." << endl;
} //try to guess the output
```

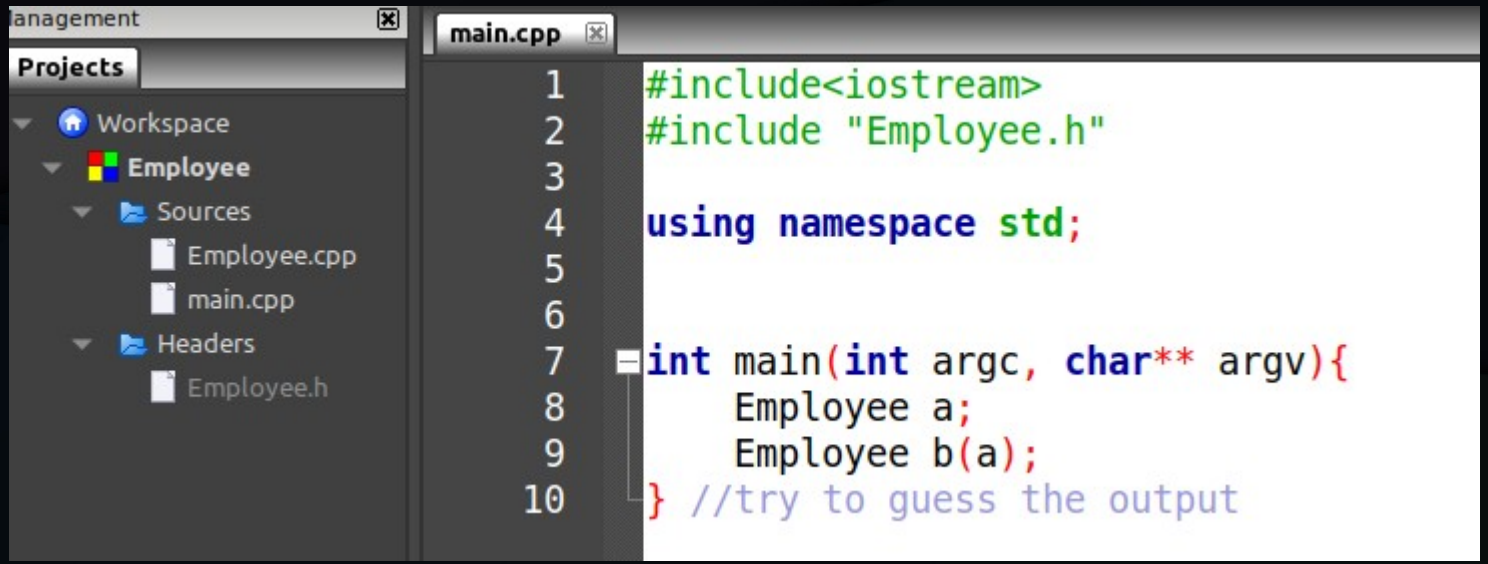
Compiling Multiple Files

- g++ can be used to compile multiple files at once.
- Only .cpp files need to be compiled.

```
gentry@lappy:~/Desktop/CS2308_Spring22_content/Lectures/Lecture Examples/Employee$ ls
Employee.cpp  Employee.h  main.cpp
gentry@lappy:~/Desktop/CS2308_Spring22_content/Lectures/Lecture Examples/Employee$ g++ Employee.cpp main.cpp -o main
gentry@lappy:~/Desktop/CS2308_Spring22_content/Lectures/Lecture Examples/Employee$ ./main
gentry@lappy:~/Desktop/CS2308_Spring22_content/Lectures/Lecture Examples/Employee$ ls
Employee.cpp  Employee.h  main  main.cpp
gentry@lappy:~/Desktop/CS2308_Spring22_content/Lectures/Lecture Examples/Employee$
```

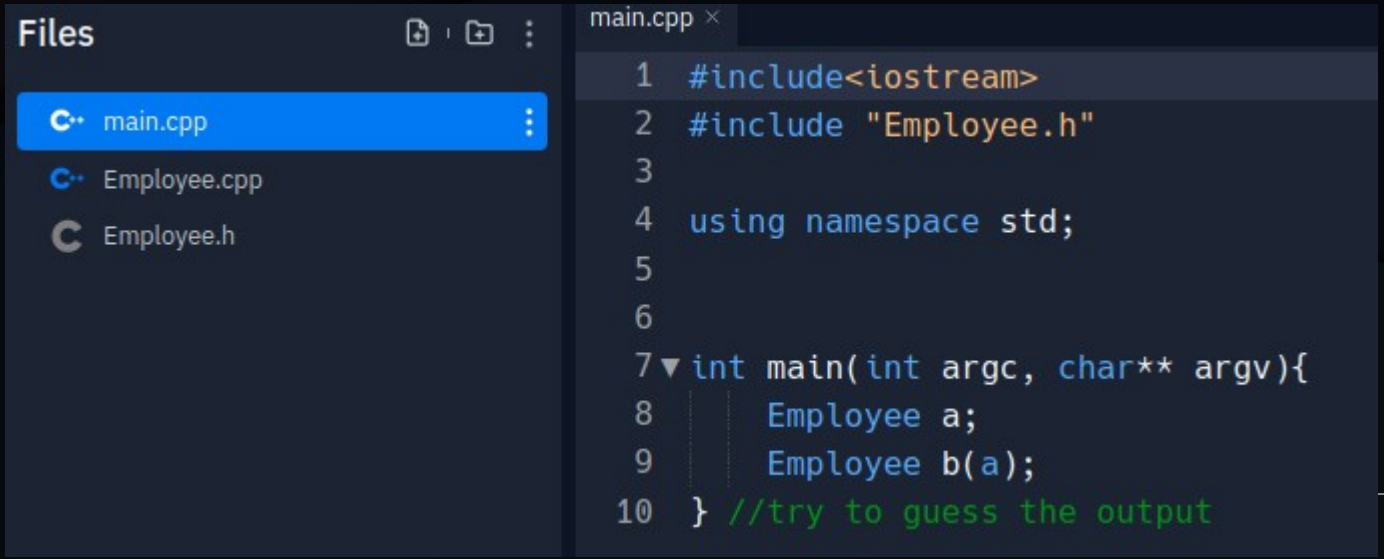
Compiling Multiple Files

- A CodeBlocks project can be used to manage multiple code files.
- Use Project → add files… to incorporate files into an empty project.



Compiling Multiple Files

- Use “Add File” to create a new blank file in Replit or “Upload File” to add a local file.
- Running a project will automatically link files.



Just Remember...

- Only one main.cpp or main function can be defined in a project.
- Never #**include** a .cpp file.
- The implementation file should #**include** the the header file
  - E.g. example.cpp should #include example.h

Questions or Comments?