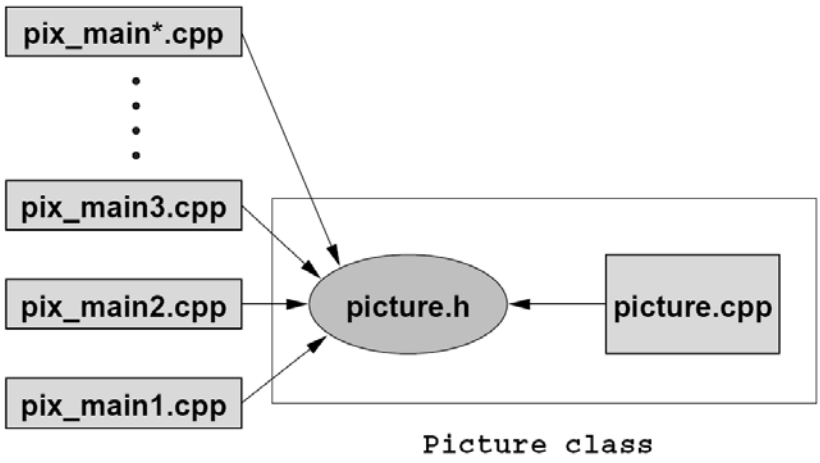
Separate Compilation?

1. Compiling a single C++ file doesn't take that long. . .
2. If we change one .c++ file, we do not want to re-compile all of them — it might take too long to complete
3. Just re-compile only the changed .c++ file to produce an object file and then link it with the other object files

Why Separate Compilation? - Encapsulation Reviewed

- Encapsulation principle:
  - Separate how class is used by programmer from details of class's implementation
- "Complete" separation
  - Change to implementation ◊ NO impact on any other programs
- Basic OOP principle



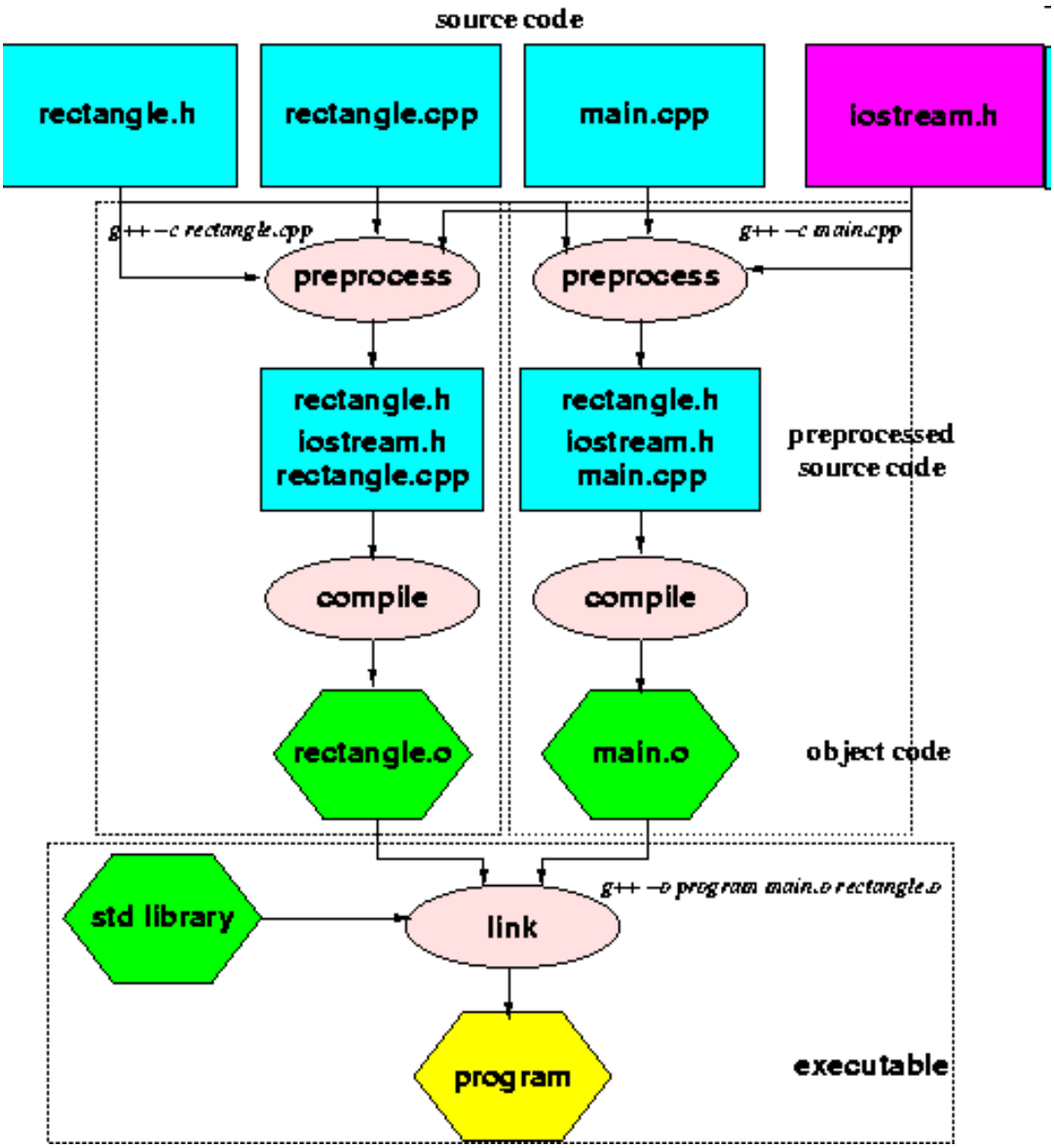
Picture class

Build library of classes

- Separate from "using" programs
- Re-used by many different programs
- Just like predefined libraries

How does it work?

- Program Parts
  - Kept in separate files
  - Compiled separately
  - Linked together before program runs
- Class Independence
  - Separate class definition/specification
    - Called "interface"
  - Separate class implementation
  - Place in two files



Class Separation

- Interface File (header, *.h)
  - Contains class definition with function and operator declarations/prototypes
  - Users "see" this, Programs that use class will "include" it
    - #include "myclass.h"
  - Separate compilation unit
- Implementation File (*.cpp)
  - Contains member function definitions
  - Separate compilation unit

Demo: dtime.h, dtime.cpp timedemo.cpp

g++ -c dtime.cpp
g++ -c timedemo.cpp
g++ dtime.o timedemo.o –o timedemo