## Demos: advanced class design

- ~mikec/cs32/demos/IntArray/ files
  - Mostly about dealing with objects pointing to dynamic memory
- ~mikec/cs32/demos/String/ files
  - Full-featured string-like class, with many overloaded operators and other functions that are not part of the textbook’s StringVar class

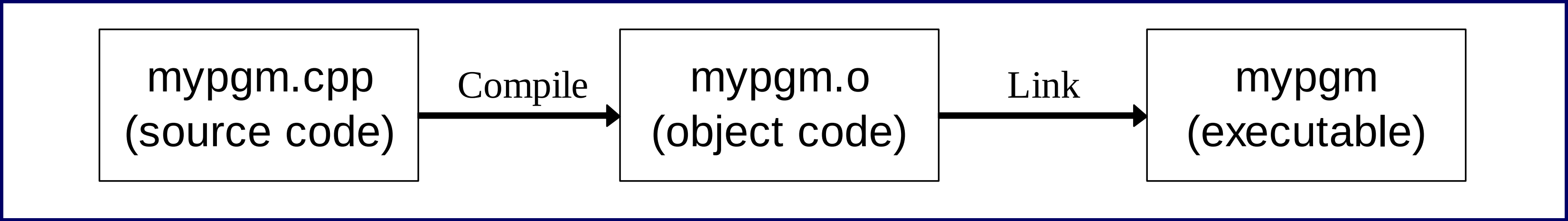
# About building a program so Linux (the OS) can run it

Starting to learn what gcc/g++ does
(learned how to use g++ in labs)

Based on Reading #5

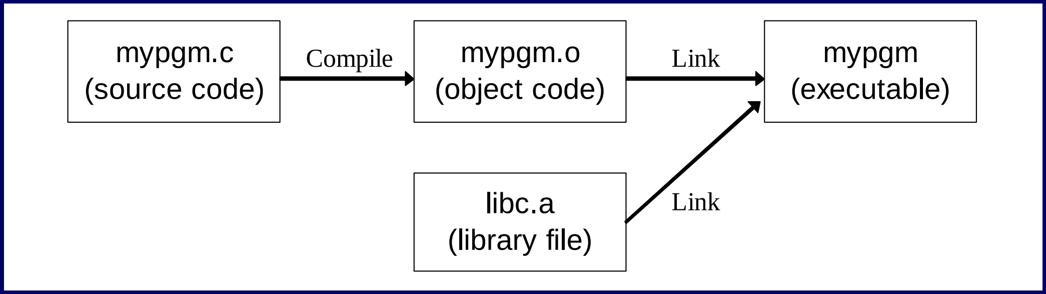
## Program building

- Have: source code – human readable instructions
- Need: machine language program – binary instructions and associated data regions, ready to be executed
- g++/gcc does two basic steps: compile, then link
  - To compile means translate to object code
  - To link  means to combine with other object code (including library code) into an executable program



# Link combines object codes

- From multiple source files and/or libraries
  - e.g., always libc.a



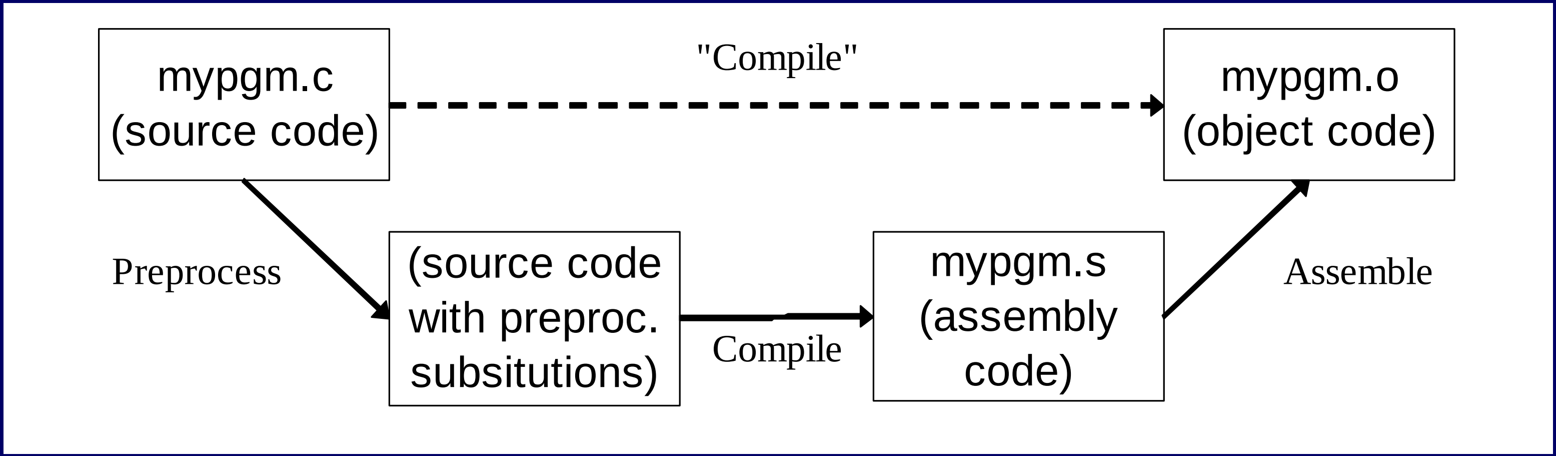
- Use -c option with gcc/g++ to stop after creating .o file

```
-bash-4.1$ gcc -c mypgm.c ; ls mypgm*
mypgm.c  mypgm.o
```

  - Is necessary to compile a file without a main function
- Later link it to libraries – alone or with other object files:

```
-bash-4.1$ gcc -o mypgm mypgm.o ; ls mypgm*
mypgm  mypgm.c  mypgm.o
```

## Compiling: 3 steps with C/C++



- First the preprocessor runs
  - Creates temporary source code with text substitutions as directed
  - Use `gcc -E` (or just `cpp`) to run it alone – output goes to `stdout`
- Then the source is actually compiled to assembly code
  - Use `gcc -S` to stop at this step and save code in `.s` file
- Last, assembler produces the object code (machine language)

## Automate builds with make
(a short follow-up to lab04)

- make is a Unix/gnu tool that executes actions as necessary to satisfy dependencies
- First create a "Makefile" (see Lab04 and hw4 for tips)

```
pgm: pgm.o                          # dependency
        gcc  pgm.o -o pgm           # action (tab required)
pgm.o: pgm.c
        gcc -c pgm.c
```

- Why bother learning, and using the make tool?
  - Some projects have many, many modules; even many programmers. Automated, so guarantees complete and up-to-date builds, without needless steps.
  - Just type "make” – the program does the rest

Second Exam

Thursday, November 8