# Lab 07: Build & Use Static Library

A library is a collection of pre-compiled object files that can be linked into your programs via the linker. Examples are the system functions such as printf() and sqrt(). There are two types of external libraries: *static library* and *shared library*.

- **A static library** has file extension of **".a"** (archive file) in Unixes or **".lib"** (library) in Windows.
- **A shared library** has file extension of **".so"** (shared objects) in Unixes or **".dll"** (dynamic link library) in Windows. Because of the advantage of dynamic linking, GCC, by default, links to the shared library if it is available. You can list the contents of a library via "nm filename".

## How to build a static library:

We use **ar** comand to maintain archive libraries in C/C++. The archive library is a collection of files, typically object files. Using **ar**, you can create a new library, add members to an existing library, delete members from a library, extract members from a library, and print a table of contents for a library.

**Example:**

Create a static library, **libsort.a,** that contains **bubbleSort(), selectionSort()** and **insertionSort()** functions which are defined in each files(~~.cpp), respectively.

Let's suppose that you have a few source files (**src/bubble.cpp, src/selection.cpp, src/insertion.cpp**) to turn it into a static library (**lib/libsort.a**).

Assume that you are currently working in **src** folder and keep include files in **~/include/sort.h**.

```
> g++ -c bubble.cpp -o bubble.o
> g++ -c selection.cpp -o selection.o

> ar rcs libsort.a bubble.o selection.o insertion.o
> ar                    // list all the options available
> ar t libsort.a        // list ~.o files archived
> ar x libsort.a insertion.o // extract ~.o files archived
> ar d libsort.a insertion.o  // delete ~.o files archived
> nm bubble.o           // list the actual function names in .o file

> cp libsort.a ../lib    // copy it and save in lib folder
```

```
ar flags:
  r: replace or insert files into archive (with replacement).
  c: create an archive file
```

```
   d: delete files in archive file
   s: regenerate the external symbol table
   t: display contents of archive
```

**NOTE:** It is important that you recognize that the GCC compiler requires that you prefix your static library with the keyword **lib** and suffix **.a**, like **lib**sort**.a**. The lib prefix is required by the linker to find the static library.

## How to reference a static library

### Recall the following flags:

```
g++ flags:
 -I: Specifies the folder name where the header files (~.h) exist.
 -L: Specifies the folder name where the referenced library exists.
 -l: Specifies the library name you want to attach (without 'lib' in its name)
     For example, use -lnowic if its file name is libnowic.a
```

### Example 1. Build an executable, sort.exe, with sortDriver.cpp and libsort.a

If you are present at ~nowic/labs/lab07 and you have **not** moved libsort.a into nowic/lib folder yet, then you can create and run an executable 'sort.exe' as shown below:

```
> g++ sortDriver.cpp print_list.cpp -o sort -I../../include -L./ -lsort
> ./sort
```

If you are present at ~nowic/labs/lab07 and you have moved libsort.a into nowic/lib folder, then you can create an executable 'sort.exe' as shown below:

```
> g++ sortDriver.cpp print_list.cpp -o sort -I../../include -L../../lib -lsort
> ./sort
```

If you are additionally using some functions in nowic.a which exists in ~nowic/lib, you may add it at the end as shown below

```
> g++ sortDriver.cpp print_list.cpp -o sort -I../../include -L../../lib -lsort -lnowic
```

### Example 2. Build an executable, nowic.exe, with nowicDriver.cpp and libnowic.a

If you are present at ~nowic/labs/lab07 and libnowic.a exists in nowic/lib folder, then you can create an executable 'nowic.exe' as shown below:

```
> g++ nowicDriver.cpp -o nowic -I../../include -L../../lib -lnowic
> ./nowic
```

## Files to submit

Use sortDriver.cpp and nowicDriver.cpp to build the following executables, respectively.  You must use the two static libraries you built in this lab.

- sort.exe
- nowic.exe

Grade: 1.0 point