Software libraries

Victor Eijkhout, Susan Lindsey

Fall 2022

last formatted: August 24, 2022

1. Don't reinvent the wheel: use a library

Many things you want to program have been thought of before:
see if there is a library for it.

Library: 'program without main':
you supply the main, functionality comes from library

2. External libraries: usage

Suppose the 'fancy' library does what you need.

1. Include a header file
2. Then use the functions defined there.

```
#include "fancylib.h"

int main() {
  x = fancyfunction(y);
}
```

3. External libraries: compile

1. Compiler needs to know where the header is:

   ```
   icpc -c yourprogram.cxx -I/usr/include/fancylib
   ```

2. You may need to link a library file:

   ```
   icpc -o yourprogram yourprogram.o \
       -L/usr/lib/fancylib -lfancy
   ```

   (not for 'header only' libraries)

4. Where to find libraries

Search . . . There is a lot of stuff on github.

1: Commandline arguments

5. Traditional commandline parsing

Code:

```
cout << "Program name: " << argv[0]
     << "\n";
for (int iarg=1; iarg<argc; iarg++)
  cout << "arg " << iarg
       << " of " << argc << ": "
       << argv[iarg] << " => "
       << atoi( argv[iarg] ) << "\n";
```

Output
[args] argcv:

```
./argcv 5 12
Program name: ./argcv
arg 1: 5 => 5
arg 2: 12 => 12
./argcv abc 3.14 foo
Program name: ./argcv
arg 1: abc => 0
arg 2: 3.14 => 3
arg 3: foo => 0
```

Can you imagine coding

```
myprog --debug -f "input.txt" -n 10,20,40
```

?

6. Example: `cxxopts`

`https://github.com/jarro2783/cxxopts`

Find the `2.2.1` release.
Use `wget` or `curl` to download straight to the class machine.
Unpack it.

7. Cmake based installation

The `cxxopts-2.2.1` directory has a file `CMakeLists.txt`

```
mkdir build
cd build
cmake -D CMAKE_INSTALL_PREFIX:PATH=${HOME}/mylibs \
    ..
make
make install
```

(This is an 'in-source' build. I don't like it: prefer to have the build directory elsewhere to keep the source untouched.)

8. Let's use this library

```
#include "cxxopts.hpp"
int main() {
  cxxopts::Options
    options("programname", "Program description");
}
```

compile

```
icpc -o program source.cpp \
   -I/path/to//cxxopts/installdir/include
```

Can you compile and run this?

9. Commandline options

```
options.add_options()
  ("h,help","usage information")
  ("n,nsize","size of the thing",
   cxxopts::value<int>()->default_value("4096"))
   // et cetera
  ;
auto result = options.parse(argc, argv);
if (result.count("help")>0) {
  std::cout << options.help() << std::endl;
  return 0;
}
int array_size = result["nsize"].as<int>();
```

Write code to test this.

Can you add more types of options?