



**COLORADO SCHOOL OF MINES
ELECTRICAL ENGINEERING & COMPUTER SCIENCE DEPARTMENT**

**CSCI-410
Elements of Computing Systems
Spring 2014**

PY-03

In this assignment, we will use the classes implemented in the previous two assignments to implement a program that you can run on your files before submitting them to ensure that the proper headers are in place. The command line format will be:

```
prompt> addHeaders (filename|directory) project
```

Notice that the command line name is NOT “py03”, but rather a more meaningful name. Your directory and ZIP file will still reflect the PY03 project, only your top level script name will be different. This is to prepare you for the ECS projects which will specify the program names you are to use.

The user will do one of the following actions:

- 1) From within the directory containing the file, run the program with the **source** argument being a filename (including extension). The program will then create a backup copy of the file named “filename_bak” (with the same extension). It will then create a new version of the original file with the header placed at the beginning and the contents of the original file following the header.
- 2) From the directory immediately above the files to be modified, run the program with the **source** argument being the directory name. The program will then work through all of the files having extensions that have specified headers and individually perform the backup and new version generation described above.

The **project** argument is the project name that is to appear in the header.

The program need not check if a file already has the proper header, however it is your responsibility to NOT turn in files that contain duplicate headers.

Note that you will need to be able to process multiple file lists, each with a different extension. With a little bit of thought, you should see how you can do this with a very simply for loop in which each iteration of the loop creates a new **FileSet** object for a particular extension.

You should also note that the user supplies a project name but does not supply other information such as ADIT UserID or other information. With the exception of the project and the filename, the contents of the header are static for a given person. Thus you can add code to the **FileSet** constructor to initialize these parameters. The filename, of course, is known at the time that each file is processed.

Be aware that you should NOT use O/S utilities, such as **cat**, to manipulate the files since that will make your program O/S specific. Instead, either use platform independent functions from the **os** module or perform the desired operations at a more primitive level.

As a recommendation, do not attempt to hammer out the entire program and then see if it works. Instead, adopt an incremental approach. Here is one possibility:

First, get your upper-level code working so that it properly generates the file lists that need to be processed. You might copy or generate some files for it to work with, both in the current working directory (the PY03 directory) and in a couple of directories below it. The directories should contain a selection of all the various file types that have defined headers (as well as a few that don't just to test whether their presence causes problems). For this stage, just having your program walk through the set of file extensions and use the **FileSet** object's **report** function should be sufficient.

For the second stage, have the program generate the new files but use the “_bak” name on these and have them contain the header only. You do not even need to open the original files at all and, while you are working out the bugs, you don't need to delete the new files and restore the names of the old files.

For the third stage, have the program copy the contents of the original file and append it to the new files (still keeping the name of the new files with the “_bak” modifier for the same reason.

At this point, you are ready to finalize your program by simply copying the contents of the original file to the “_bak” version and then creating the new version with the original name and copying the contents from the “_bak” version back into the new version.