Homework Companion
Software Engineering Concepts (ECE473-573)

Jonathan Sprinkle

Spring, 2011

http://www.ece.arizona.edu/~ece473/

Document Description:

This Homework Companion is designed to reduce the ambiguity in grading, and to provide you and the instructors with a basis for objectifying the assignment of grades. In the event of any ambiguity, however, the instructors' decision will be authoritative.

Homework Requirements

All homework assignments must be submitted via the `d2l` program. Files must be submitted according to the following specifications. Failure to obey all of these guidelines may result in a failing grade, regardless of the correctness of your program!!

- The homework must be submitted using `d2l`. The submitted file should be a `.tgz` file, named `hwNN-lastname.tgz`*.
- No paper copy of the source code is required as part of your submission, unless specified as part of the assignment.
- Any questions that require a written (i.e., English language) answer should be contained in a *single* text file, PDF document, or MS Word document in the root directory of the tar file.
- Source code submissions
  - All files required to compile the submitted source code should be included in the `.tgz` file in the appropriate directory structure. You **should** include project files when submitting, but not compiled files.
- Effect of submission mistakes on homework grades:
  - Code does not compile—up to 40 point reduction
  - Code will not link—up to 30 point reduction
- Other considerations
  - Submissions issues
    - Failure to arrive before deadline —considered late
    - Submitted to incorrect `d2l` assignment—considered late, until correctly submitted

*If you share the last name of someone else in the course, you should use `hwNN-lastname-firstname.tgz` as your filename.

    - Submitter name not appropriately given or placed *in every submitted file*—up to 100 point reduction, based on instructors' discretion
    - Homework archive incorrectly named or created–up to 100 point reduction, based on instructors' discretion
    - Inclusion of compiled or generated project files in submission archive—up to 10% reduction
  - Code is not documented—up to 50% reduction of already discounted grade
  - Memory leak(s)—automatic 10% reduction of already discounted grade
  - Code does not solve required problem—up to 100 point reduction, based on instructors' discretion

Specific reduction details

Note that all reductions are *compounding*, i.e., code that does not compile definitely does not run, so an automatic 70 points are lost (meaning that the grading of the assignment starts at 30). If that code is also undocumented, then a final grade of 18/100 will be discounted to 9/100.

Failure to satisfy unspecified requirements

Deductions may be taken, at the discretion of the instructor, due to failure to satisfy requirements that were not specified by the homework assignment. Specific examples of this from previous terms include:

- Failing to use APIs effectively;
- Saving a file to a restricted area of data storage, rather than following best practices;
- Using filenames that fail to convey the purpose of the file, or are unlikely to be unique;
- Selecting a menu flow that is unnecessarily complex;
- Failing to fully account for and consider issues of app selection from the home screen, startup, task switching, or shutdown;
- Providing a non-robust implementation that consistently exhibits strange behavior under certain circumstances or usage patterns; and
- Exhibiting slow response, due to poor coding practices or inferior design.

In short, deductions may be taken if the instructor deems that someone might be discouraged from selecting this solution over another solution in the "real world." This may mean that your solution, while meeting all specified requirements, and technically correct, will receive a grade much less than you expect. **This is due to the fact that this course is the capstone software course, and expectations are higher than for foundational programming and design courses.** Your solutions are expected to be fully rounded, not just a minimal implementation.

Late Submission

Start your assignments early. Unless otherwise noted, all homework assignments are due at 11:59 PM on the day specified. An exponential decay will be applied to the grade: for each 0-24 hour period it is late, the final grade will be multiplied by 0.63, until the graded homework assignments of other students are returned. At the return of the homework, the assigned grade of un-submitted homework assignments will be 0. Unless per-homework accommodations are made in advance, no exceptions to this policy will be permitted, and no makeups will be allowed.

Un-runnable Code

Code does not compile:

- 40 point reduction if the code will not compile at all
- 30 point reduction if the code will compile after instructor tweaking[†]

Code will not link[‡]:

- 30 point reduction if the code does not run or link
- 25 point reduction if code runs after instructor tweaking[§]

Undocumented Code

All code submissions must be commented. If you do not comment your code sufficiently, you will have points removed as described above. You may (or may not) receive written comments recorded on the printout of your code "text." *Do not forget to include your name as part of the documentation!*

Contents of homework archive

Your homework archive must be named `hwNN-lastname.tgz`, where `NN` is the homework number (defined in each homework document) and `lastname` is your lastname[¶]. When your `.tgz` file is extracted, there should be no files in the root, only a directory that *must be named* ***hwNN-lastname***. Failure to correctly name your homework archive, or to have all contents contained in an appropriately named directory, will be catastrophic to your grade for that submission.

Please confirm that you are NOT including files such as `._DS_Store`, or `Filename.cpp~` (note the trailing `~`) in your submission. You can have a command line interface "tell" the contents of your tar file by typing the following in a Bash shell (such as `bracton.ece.arizona.edu`):

```
$ tar tzf hwNN-lastname.tgz
```

When submitting the `.tgz` archive of your homework assignment, you should include *only* source files, unless instructed otherwise. Recall that if you include *binary* files (generated by the compile process) in your homework submission, you may receive up to a 10% reduction. These binaries include the various files created in some build directory that your IDE uses.

In order to easily name your directory, the appropriate structure is recommended for your homework assignments:

```
$ ls
hw01-lastname    hw02-lastname    hw03-lastname    hw04-lastname
hw05-lastname    hw06-lastname    hw07-lastname
$ cd hw01-lastname
$ ls
README.txt src              build
```

This is now an easy way to produce your homework `.tgz` file:

```
$ pwd
/home/username/ece473/src/homeworks/hw01-lastname
$ ls
README.txt src              build
```

[†]Instructor reserves the right to correct non-compiling code, but is not guaranteed to do so.
[‡]All code that does not compile *automatically* does not link
[§]Instructor reserves the right to correct non-running code, but is not guaranteed to do so.
[¶]If you share the last name of someone else in the course, you should use `hwNN-lastname-firstname.tgz` as your filename

```
$ rm -rf build
$ ls
README.txt src
$ cd ..
$ tar czf hw01-lastname.tgz hw01-lastname
```

Making good code structure habits such as this will pay off in your future.

Supplements to the Homework Companion

Individual assignments may have requirements which differ from the Homework Companion. In those cases, the assignment's specification will take precedence over the Homework Companion, and the resolution of any ambiguities will be decided by the instructor(s).

Policy for Office Hour Questions after `hw02`

After the submission of `hw02`, the instructor(s) reserve the right to decline to answer specific questions about the homework on the day before it is due. E.g., if a homework is due at 11:59 PM on a Wednesday, then any Tuesday consultations of that week would consist *only* of homework-independent clarifying questions related to the course or to programming. It would be appropriate to ask, for example, "Is [this idea] correct?" and *inappropriate* to come in and say "I am having trouble understanding the requirements of problem 2 for `[hwNN]`"

This policy is intended to motivate you to ask questions and begin the homework assignments early. Specific homework questions can still be asked before class if the instructor permits.

Reminder: Academic Integrity

Students are expected to do all work by themselves, except when specified by the instructor in writing. This includes using source code from the internet, as well as code you may receive from students who have previously taken this course. **If you refer to someone else's solution(s) to a homework assignment while preparing a homework submission for this course, then you are cheating—even if the solution is from a previous offering of this course.** Automatic grading takes advantage of the Measure of Software Similarity (MOSS) Project (see http://theory.stanford.edu/~aiken/moss/).

Do not work in teams on your homework—this not only violates the academic integrity policy for the university and the course, but it also penalizes projects by allowing team members to progress in the class without learning fundamentals. You may help each other when performing program assignments, but do not share code (even by showing someone how you did it), and do not write code for any member of the class (not even to fix a compile problem quickly). You may describe your approach, or create examples distinct from the homework, to help another student understand a concept, but generally this is the purpose of office hours with the instructor.