Homework Submission Guidelines*

Jeffrey D. Oldham

2000 Mar 17

Revisions:

- 2000Mar17: Added Section 2.1 about creating a `.forward` file which will route the automatic reply for homework submissions to your Trinity emailbox.
- 2000Jan26: Added section specifying use of `g++ -Wall -pedantic` when grading homeworks.

As mentioned in the syllabus, homework will be submitted via email. Please consider following these guidelines so our grading programs do not break on your code.

1 What to Submit

Your program's source code, submitted electronically, as described below. Every file must begin with the following:

- Comments giving your name and the assignment number.
- Comments documenting any choices or assumptions you made in writing the program. For example, if input is a sequence of numbers, and you assume there will never be more than 100, you should mention this assumption in the opening comments.

If your program consists of multiple source files, submit a makefile to recreate your executable program by typing `make`. Makefiles will be discussed in class.

Always try to turn in something even if it's not completely correct or complete. Partial credit will be given when appropriate.

2 Submitting Programs via Email

To submit your program, e-mail its source code to the address corresponding to your section number: cs1321-1@cs.trinity.edu or cs1321-2@cs.trinity.edu.

To e-mail source code, use one of the following approaches.

- From Linux/Unix, type the following command:
  `shar -m` *`sourcefiles`* `| mail -s "CS1321 Homework` *`N`*`" cs1321-`*`n`*`@cs.trinity.edu`

  where

  sourcefiles is a list of files containing your source code.

  N specifies the homework number, e.g., 1.

  n is your section number, i.e., 1 xor 2.

  If the message cannot be delivered, you will get mail to that effect, but the mail will be sent to you at the machine where you issued the mail command. To have mail forwarded to you at another address (e.g., the one where you usually read mail), create a text file called ".forward" in your home directory containing the address to which you want mail forwarded. (For example, if you want mail forwarded to your TUCC account, this file should contain the single line of text *yourname*@trinity.edu.)

- Using any mail program that supports attachments, send a message to the appropriate address (described above), with a subject header as specified for the assignment, and attach the file(s) containing the source code. The result should be a single mail message with one or more attachments, one for each source code file.

If for some reason you submit a homework and then want to submit it again, just repeat the above procedure. I will grade only the most recent version I receive.

2.1 Automatic Reply to Homework Submission

After submitting homework to the appropriate email address, you should receive an automatic reply indicating the homework was received. This reply is sent to the computer used to submit your homework. For example, if you use a computer science computer to submit, the reply will be sent there.

To have all email sent to any computer science computer automatically forwarded to another account, use the following procedure:

1. In your CS account's home directory, create a file called `.forward`, with exactly one line listing the email address where you want your email forwarded. This is probably your TUCC email address, e.g., `Foo.Bar@trinity.edu`.

2. Make the file readable using `chmod a+r .forward`.

To check the contents of the file, type `cat .forward`.

3 How We Will Grade the Programs

We will grade the programs using one of the Trinity Computer Science machines available to you. (I believe the operating systems are all the same on all machines available to you.)

In particular, we will be compiling using commands like

```
g++ -Wall -pedantic foo.cc -o foo
```

These compilations should not produce any errors or warnings. (Warnings caused by incorrectly configured system files will be ignored, but I have run across very few of these.) Using the `-Wall -pedantic` options helps ensure your code is more likely to be portable to any machine.

Note that we will be grading using `g++` on the Linux operating, not Visual C++.