

Advanced Object-Oriented Programming, Spring 2018

Homework Assignment #2

Due midnight Wednesday, April 18, 2017

Instructions

1. If any question is unclear, please ask for a clarification.
2. You may try to reuse as much of the source code supplemented as possible.
3. Unless stated otherwise, all the line numbers for the program listings are for reference only.
4. You are required to do all the homework assignments on Linux and use g++ version 4.9.2 or later.
5. You are required to give your TA a demo of your program. Make sure that your program can compile and run on the server machine, which will be used for the demo.
6. For the program that you write, you are required to include a Makefile. Otherwise, your homework will not be graded—meaning that you will receive zero marks.
7. Unless stated otherwise, you are required to work on the homework assignment individually.
8. No late homework will be accepted.

Programming Project

This assignment requires that you write a program in C++ to list the contents of a **TAR** file. The format of TAR files can be found at http://en.wikipedia.org/wiki/Tar_file_format.

Here is a sample run of your program, assuming the name of your program is **mytar**.

```
$ ./mytar /student/test.tar
total 8 files
-rw-r--r-- root          root          512 b9134001/512byte
drwxr-xr-x root          wheel          0 b9134001/hw1/
-rw-r--r-- root          wheel        256 b9134001/hw1/Makefile
-rw-r--r-- root          wheel        551 b9134001/hw1/tarfile.h
-rw-r--r-- root          wheel        318 b9134001/hw1/main.cpp
-rw-r--r-- root          wheel        148 b9134001/hw1/main.h
```

```

-rw-r--r-- root          wheel          1662 b9134001/hw1/tarfile.cpp
drwx----- b9134001     student        0 b9134001/hw3/
$ ./mytar mytar
Not a ustar file.
File opening error.
$

```

The output of your program has to be similar to that of the **tar** command. Also, your program has to check to see if the file type is “USTAR format” as described in the document. The test data **test.tar** can be found in the directory **/student**. Note that this is also the file to be used by your TA to test your program during the demo.

You can use the structure given below to read the file header.

```

struct TarHeader {
    char filename[100];
    char filemode[8];
    char userid[8];
    char groupid[8];
    char filesize[12];
    char mtime[12];
    char checksum[8];
    char type;
    char lname[100];

    /* USTAR Section */
    char USTAR_id[6];
    char USTAR_ver[2];
    char username[32];
    char groupname[32];
    char devmajor[8];
    char devminor[8];
    char prefix[155];
    char pad[12];
};

```

The `seekg()` member function of `iostream` provides random access capability. For example, the call `seekg(5, ios::beg)` sets the read pointer to the fifth byte of the file while the call `seekg(8, ios::cur)` moves the current pointer forward 8 bytes.

You are also required to use the `read()` function instead of the “>>” operator—to make it easier to control your program.

Grading Policy

The grading policy for this assignment is as follows:

- Make sure that a **Makefile**, which contains at least three targets—**all**, **dep**, and **clean**—is provided. Otherwise, the grade for your program will be zero.
- 50 points if your program compiles without errors and warnings, and the answer is correct.
- 10 points if your program is modularized and has at least two .cpp files and one .h file. For instance, insofar as this homework assignment is concerned, you have **main.cpp**, **mytar.h**, and **mytar.cpp**.
- 10 points if your program is well-structured.
- 10 points if the main function contains less than 15 lines of code, and you don't put more than one statement in a line.
- 20 points if you use only C++ Streams and File I/O—as given in Chapter 12 of the text—for all the I/Os. In other words, *no* C library function calls such as **open**, **close**, **read**, **write**, the **scanf** family of functions, the **printf** family of functions, are used in your program.