WRITEUP

csclipei

1. Testing

   The testing I did for my program followed the sequential outline of my program. I first checked to see if the user's only argument inputted was the program call itself, which would then print back everything the user inputted to stdout. The second test I did was to check if the user inputted a file name, which would then open the file and if the return value of the file descriptor is valid, would read the contents of the file and the write back to the stdout. I also checked if the user inputted a file that didn't exist in which case the program would print an error message and continue to the next file, if there is one. Then the remaining tests I did handling the '-' character which would prompt users to input and write back to the stdout whatever the typed. If the user inputted Ctrl+D to signal EOF, the program would either end or move on to the next file if there is one.

2. Answer

   The code for handling file differs form that for handling standard input in that when the code opens a file from the standard input it returns a file descriptor which is then used to read the file its opened. Whereas, handling standard input was just using a value of 0 for read which meant reading from standard input. This is an example of modularity since you are breaking down the fundamentals of *dog* by opening a file – if there exists one – reading a file or the users input, writing to the standard output and then finally closing the file. This program is comprised of components that encapsulates the purpose of the code which should mimic *cat*'s basic functions.