1. Drew Sadler
2. #include<sys/types.h>
   #include<sys/stat.h>
   #include<fcntl.h>
   #include<unistd.h>
   #include<stdio.h>
   #include<stdlib.h>
   #define bufferSize 1200
   int main(int argc, char* argv[])
   {
   char* filename = argv[1];
   char buffer[bufferSize];
   int fd = open(filename,O_RDONLY);

   while(read(fd,buffer,sizeof(fd)-3)!=0)
   {
   printf(buffer);
   }

   close(fd);
   return 0;
   }
3. Runs continuously for infinity
4. Breaks and runs for infinity as well
5. -1 if an error occurred
6. if(fd==-1)
   {
   exit(-1);
   }
7. perror("Error opening and reading file: ");
8.
9. Prints out my error message of
   Error opening and reading file: : No such file or directory

10. Prints and raises the error message again
    Error opening and reading file: : Permission denied

11. Open(): -1 is returned and *errno* is set to indicate the error
    Close():-1 is returned, and *errno* is set to indicate the error.
    Read():-1 is returned, and *errno* is set to indicate the error
    Write():-1 is returned, and *errno* is set to indicate the error
    Printf();If an output error is encountered, a negative value is returned. But no errno is set.

12. Error checking simplifies design due to not having to make and configure more variables to get around issues, and can sort of force the user to use the program the way developers want through calling errors. Also it allows developers to see that certain functions return certain values or emplace certain functions when failing, and can use the data of what elements come from the fails to stop the program from crashing or route it differently when encountering a failure.