

- How does the code for handling a file differs from that for handling standard input? What concept is this an example of?

To handle files you pass a file descriptor that is returned from open to read's fd. To read from std in, you pass 0 into read's fd. This is a concept of abstraction/modularity.

Testing

For testing I ran this program on no files, some files, and a mix of some files and -'s. Files can include text, binary, or directory files. I then compared dog results with cat. To make life easier I piped all input/output of cat and dog into separate files and ran diff on them to compare the differences. Tested on files up to 2GB including big text or binary files.

Example runs:

```
./dog Makefile dog.cpp - Makefile
./dog - Makefile dog.cpp Makefile
./dog -Makefile dog.cpp
./dog - Makefile dog.cpp
```