File I/O with Text Files

Reading to files and writing from files is very useful.

Basically, every file operation is:

- Open your file
- Read or Write your file
- Close your file

So

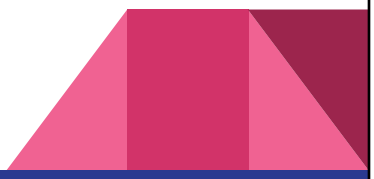
- First we need a file pointer to point to the file:
  - `FILE *fp;  // declare FILE * variable`
- Second we open the file for mode reading, writing or appending ("r", "w")
  - `fp = fopen("myfile.txt", "r")`
  - r: reading – w: writing (erases the file!)
- When finished use:
  - `fclose(fp);`

Reading and Writing

- Instead of printf, just use fprintf
  - `fprintf(fptr, "%s %d", firstName, iAge);`
- Instead of scanf just use fscanf
  - `fscanf(fptr, "%s %d", &firstName, &iAge);`
- We can easily create, view and edit our files in Notepad.

List of file commands

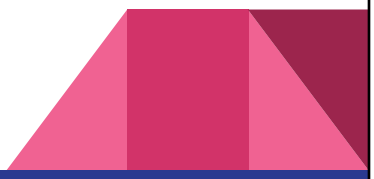
- fopen (open the file)
- fclose (close the file)
- fprintf (use printf notation to write text to file)
- fscanf (use scanf notation to read the file)
- feof (have we "hit the End Of File" yet?)
- fgets (a different way to read text from the file)
- fputs (a different way to write text to the file)



The file pointer (cursor)

When reading from or writing to a file,
imagine the cursor moving through the file.

Every read or write operation moves this cursor through the file.



Example: Writing name to a file

```
int main() {
    FILE *fptr;
    char name[50];

   fptr = fopen("name.txt", “w");

   printf("Enter name: ");
   scanf("%s", name);

   fprintf(fptr, "%s\n", name);
   fclose(fptr);

    return(0);
}
```

Example: Reading name from a file

```
int main() {
    FILE *fptr;
    char name[50];

   fptr = fopen("name.txt", "r");

   fscanf(fptr, "%s", name);

   printf("Your Name is %s.",name);

   fclose(fptr);

   return(0);
}
```

Example: Reading unlimited number of names from a file

```
int main() {
    FILE *fptr;
    char name[50];

    fptr = fopen("name.txt", "r");

    while (!feof(fp)) {
        fscanf(fptr, "%s", name);
        printf("Name is %s \n",name);
    }

    fclose(fptr);

    return(0);
}
```

In bigger programs…

- Use files to populate your arrays of data, but **avoid the trap of always writing to or reading from the file.**
- The general flow should be:
  - Start of program
  - Read from file into your variables (and close the file)
  - Run main program (menu, etc, etc)
  - Quit?
    - Write to the file by erasing and re-creating the file from the data in your variables

DO NOT READ OR WRITE IN THE MIDDLE OF YOUR PROGRAM

Menu program example

```c
#include <stdio.h>

int main() {
  FILE *fp;
  int num[1000];
  int x,numNumbers=0,choice;

  // READ DATA IN
  fp = fopen("numbers.txt","r");
  while (!feof(fp)) {
    fscanf(fp,"%i",&num[numNumbers]);

    if (!feof(fp))
      numNumbers++;
  }
  fclose(fp);

  //RUN MAIN PROGRAM (*** NO FILE I/O IN HERE ***)
  do {
    system("clear");
    printf("-------\n");
    for (x=0;x<numNumbers;x++) {
      printf("[%i] %i\n",x,num[x]);
    }
    printf("-------\n");

    printf("1. Add number\n");
    printf("0. Quit\n");
    printf("==> ");
    scanf("%i",&choice);
    getchar();

    if (choice == 1) {  //ADD
      printf("Enter a new number: ");
      scanf("%i",&num[numNumbers]);
      getchar();
      numNumbers += 1;
    }

    else if (choice != 0) {
      printf("\nInvalid Choice! Please try again
    }

    if (choice != 0) {
      printf("Press enter to continue...");
      getchar();
    }
  } while (choice != 0);

  //QUIT, SAVE TO FILE
  fp = fopen("numbers.txt","w");
  for (x=0;x<numNumbers;x++) {
    fprintf(fp,"%i\n",num[x]);
  }
  fclose(fp);
```