

Announcements

Thursday Extras completed for semester:

- If you have feedback on the candidates (either 4:15 talk or Wednesday's class), be sure to send me your comments by Sunday at 5:00 pm

Office Hours:

Today: 1:30 - 2:30, 4:00-5:00 (different than usual, since I have a meeting with the Dean)

I'll be back in the office, whenever the meeting is over.

Next week: as normal (I hope)

Extra Credit Supplemental Problems:

- To be considered for this week, they must be received (in paper and electronic form) by 5:00 pm this afternoon.
- No more than 2 may be submitted between this afternoon at 5:00 pm and Friday, December 13 at 5:00 pm
- Absolute deadline: Friday, December 13 at 5:00 pm; anything after this time is not accepted!!

Personal Care!!!!

Take care of yourself: meals, sleep, hygiene (e.g., be sure to shower, wash clothes)

End-of-course Evaluations: Please complete online soon!

Today: File input/output

Monday

New lab partners

Robot Parade: getting started

Wednesday: Robot Parade: working toward completion

Thursday: Office hours (no sign-ups): 2:30-4:00, 5:00-7:00

Friday: Robot Parade in class

Files

Main memory:

- storage/retrieval fast
- capacity limited
- deleted when program finishes
- data from user limited by typing

Files

- storage/retrieval not fast
- capacity vast
- maintained over long periods
- data gathered from many sources:
 - databases
 - sensors,
 - "internet of things": keep track of appliance activity, personal activities, Web requests (typed URLs), locations (GPS, card to unlock doors), etc.
- primary source of data for today's processing: files

Files

Suppose one wants to read an integer from a file "integer.txt" into a variable i.

Which of the following code snippets accomplishes this task?

Approach 1:

```
int i;  
fscanf ("integer.txt", "%d", &i);
```

Approach 2:

```
int i;  
FILE * infile = fopen ("integer.txt", "r");  
fscanf (infile, "%d", &i);
```

Approach 3:

```
char line [80];  
FILE * infile = fopen ("integer.txt", "r");  
fgets (line, 80, infile);  
int i = atoi (line);
```

- A. 1 only
- B. 2 only
- C. 3 only
- D. 2 or 3 (not 1)
- E. 1, 2, and 3

Consider 2 proposed ways to read an integer `intg` and a double `dbl` from the terminal:

One: `scanf ("%d %lf", &intg, &dbl);`

Two: `fscanf (stdin, "%d %lf", &intg, &dbl);`

Consider the following statements:

1. One and two functionally perform the same task.
2. The compiler translates One to Two, so they are identical behind the scenes.
3. Two requires typing 8 more characters in the C statement (including a comma and a space)

Which of these statements are true?

- A. 1, 2, and 3
- B. 1 and 2 only
- C. 2 and 3 only
- D. 3 only
- E. something else

Suppose a program opens a file for writing with the statement:

```
FILE * outfile = fopen ("output.txt", "w");
```

Suppose later the program contains one or more `fprintf` lines, such as

```
fprintf (outfile, "%d", 123456);
```

What result might occur if the program omitted `fclose (outfile)` ?

- A. The compiler will generate a warning or error.
- B. When run, the program likely stops prematurely with an error (e.g., segmentation fault).
- C. An error is likely to arise when the program is run.
- D. Some data intended to be written to the file may be lost
- E. The program likely will work fine.

Early program to convert quarts to liters:

```
#include <stdio.h>
```

```
int main ()
{
    printf ("This program converts a number of quarts to liters\n");

    int quarts;
    double liters;
    printf ("Enter integer value of quarts: ");
    scanf ("%d", &quarts);

    liters = quarts / 1.056710 ;

    printf ("%d quarts = %lf liters\n", quarts, liters);

    return 0;
}
```

Change program

- if no command-line argument, read from user
- if command-line argument, use it as file name for reading