

Scope

Scope tells us which
names are visible/
accessible

```

#include <stdlib.h>
#include <stdio.h>
#include "scanner.h"
#include "tweetHelper.h"

int getInput()
{
    printf("Enter 1 to display all tweets, enter 2 to find tweets in a certain month, enter 3 to find latest tweet\n");
    int input;
    scanf("%d", &input);

    // Must return input to use anywhere else
    return input;
}

int countRecords(char *filename)
{
    char *line;
    int count = 0;
    FILE *fp;

    fp = fopen(filename, "r");
    if (fp == 0)
    {
        fprintf(stderr, "file %s could not be opened for reading\n", filename);
        exit(1);
    }

    line = readLine(fp);
    while (!feof(fp))
    {
        count++;
        free(line);
        line = readLine(fp);
    }
    return count;
}

```

Global Scope =
not enclosed by another scope

```
#include <stdlib.h>
#include <stdio.h>
#include "sec4tweetHelper.h"

int maxMonth = 12;

int
main(int argc, char **argv)
{
    if (argc != 2)
    {
        printf("Wrong number of arguments!\n");
        exit(1);
    }

    int count = countRecords(argv[1]);
    Tweet **tweets = malloc(sizeof(Tweet *) * count);
    fillTweets(argv[1], tweets);

    int input = getInput();
    if (input == 1) displayAllTweets(tweets, count);
    else if (input == 2)
    {
        int month;
        printf("Enter a month:");
        scanf("%d", &month);
        if (month > maxMonth)
        {
            printf("Not a valid month\n");
            exit(1);
        }
        findMonthTweets(tweets, count, month);
    }
    else if (input == 3) findLatestTweet(tweets, count);
    else printf("Try another option\n");

    return 0;
}
```

```
~
~
~
```

```
#include <stdlib.h>
#include <stdio.h>
#include "sec4tweetHelper.h"
```

```
int maxMonth = 12;
```

```
int
```

```
main(int argc, char **argv)
```

```
{
```

```
    if (argc != 2)
```

```
    {
```

```
        printf("Wrong number of arguments!\n");
```

```
        exit(1);
```

```
    }
```

```
    int count = countRecords(argv[1]);
```

```
    Tweet **tweets = malloc(sizeof(Tweet *) * count);
```

```
    fillTweets(argv[1], tweets);
```

```
    int input = getInput();
```

```
    if (input == 1) displayAllTweets(tweets, count);
```

```
    else if (input == 2)
```

```
    {
```

```
        int month;
```

```
        printf("Enter a month:");
```

```
        scanf("%d", &month);
```

```
        if (month > maxMonth)
```

```
        {
```

```
            printf("Not a valid month\n");
```

```
            exit(1);
```

```
        }
```

```
        findMonthTweets(tweets, count, month);
```

```
    }
```

```
    else if (input == 3) findLatestTweet(tweets, count);
```

```
    else printf("Try another option\n");
```

```
    return 0;
```

```
}
```

```
~
~
~
```

Local to main function

Global scope
encloses main's scope

Can see **out** to enclosing scope

Cannot see **in** to enclosed
scope

Local to else if statement

```
#include <stdlib.h>
#include <stdio.h>
#include "sec4tweetHelper.h"

int maxMonth = 12;

int
main(int argc, char **argv)
{
    if (argc != 2)
    {
        printf("Wrong number of arguments!\n");
        exit(1);
    }

    int count = countRecords(argv[1]);
    Tweet **tweets = malloc(sizeof(Tweet *) * count);
    fillTweets(argv[1], tweets);

    int input = getInput();
    if (input == 1) displayAllTweets(tweets, count);
    else if (input == 2)
    {
        int month;
        printf("Enter a month:");
        scanf("%d", &month);
        if (month > maxMonth)
        {
            printf("Not a valid month\n");
            exit(1);
        }
        findMonthTweets(tweets, count, month);
    }
    else if (input == 3) findLatestTweet(tweets, count);
    else printf("Try another option\n");

    return 0;
}
```

main's scope
encloses if else's scope

```
#include <stdlib.h>
#include <stdio.h>
#include "scanner.h"
#include "tweetHelper.h"
```

```
int getInput()
{
    printf("Enter 1 to display all tweets, enter 2 to find tweets in a certain month, enter 3 to find latest tweet\n");
    int input;
    scanf("%d", &input);

    // Must return input to use anywhere else
    return input;
}
```

```
int countRecords(char *filename)
{
    char *line;
    int count = 0;
    FILE *fp;

    fp = fopen(filename, "r");
    if (fp == 0)
    {
        fprintf(stderr, "file %s could not be opened for reading\n", filename);
        exit(1);
    }

    line = readLine(fp);
    while (!feof(fp))
    {
        count++;
        free(line);
        line = readLine(fp);
    }
    return count;
}
```

Parallel scopes

Variables are local to each function and function's scopes do not enclose each other, so cannot see variables

This a will be printed

```
int a = 4;
```

```
void
```

```
epsilon(int a)
```

```
{
```

```
    printf("a is %d\n",a);
```

```
    return;
```

```
}
```

```
...
```

```
epsilon(13);
```

```
...
```

`#include` puts all names in
global scope (global
variables + function names)
from external file into your file

makefiles