CSC209 Week 4 Notes

Hyungmo Gu

May 12, 2020

Introduction to arrays in C 1 of 3

- Array
  - **Syntax:** <TYPE >VAR_NAME[ARRAY_SIZE]

```c
#include <stdio.h>

int main() {
    float daytime_high[4];
}

```

Introduction to arrays in C 2 of 3

- Accessing Array Elements
  - C doesn't check if an array access is within the bounds of array
  - Overwrites memory location if exists

```c
#include <stdio.h>

int main() {
    float daytime_high[4] = {1,2,3};
    daytime_high[5] = 999;
}

```

  - Segmentation fault occurs if suitable memory location doesn't exist.

```c
#include <stdio.h>

int main() {
    int daytime_high[4] = {1,2,3};
    daytime_high[3000] = 999;
}

```

Introduction to arrays in C 3 of 3

- Iterating Over Arrays
  - For loop
    - '<' is used over '<=' for the end condition,i.e. i <4 in for ($i = 0$; i <4; i++).

```c
#include <stdio.h>

int main() {
    float daytime_high[4] = {16.0, 12.8, 14.6, 19.1};

    float average_temp = 0;

    int i;
    for (i = 0; i < 4; i++) {
        printf("Adding element %d with value %f.\n", index
, daytime_high[i]);
        average_temp += daytime_high[i];
    }

    average_temp = average_temp / 4;
    printf("average %f\n", average_temp);

    return 0;
}

```

  - Constants
    - Combines multiple repeating values into one
    - Used to increase maintainability and readibility

```c
#include <stdio.h>
#define DAYS 4 // <-- HERE!!

int main() {
    float daytime_high[DAYS] = {16.0, 12.8, 14.6, 19.1};

    float average_temp = 0;

    int i;
    for (i = 0; i < DAYS; i++) {
        printf("Adding element %d with value %f.\n", index
, daytime_high[i]);
        average_temp += daytime_high[i];
    }

    average_temp = average_temp / DAY;
    printf("average %f\n", average_temp);

    return 0;
}

```