Why use array

Repeat repetitive tasks

Author: Alan Chu (ualch9@gmail.com)

```c
#include <time.h>   /* for time   */
#include <stdio.h>  /* for printf */
#include <stdlib.h> /* for rand   */

int random_integer(int min, int max) {
    return rand() % max + min;
}

int roll_die(void) {
    return random_integer(1, 6);
}

int main(void) {
    srand((unsigned int)time(NULL));

    int die1 = roll_die();
    int die2 = roll_die();
    int die3 = roll_die();
    int die4 = roll_die();
    int die5 = roll_die();

    // Print dice face value
    printf("Dice 1: %d", die1);
    printf("Dice 2: %d", die2);
    printf("Dice 3: %d", die3);
    printf("Dice 4: %d", die4);
    printf("Dice 5: %d", die5);

    return 0;
}
```

Oops, made a mistake.
I forgot to do newlines

Now I have to find every instance of printf and add a newline to each one

Author: Alan Chu (ualch9@gmail.com)

```c
#define kNUMBER_OF_DICE 5

int main(void) {
    srand((unsigned int)time(NULL));
    int dice[kNUMBER_OF_DICE];

    // Roll dice.
    for (int i = 0; i < kNUMBER_OF_DICE; i++) {
        dice[i] = roll_die();
    }

    // Print face values.
    for (int i = 0; i < kNUMBER_OF_DICE; i++) {
        printf("Dice %d: %d\n", i + 1, dice[i]);
    }

    return 0;
}
```

Now, if I make a mistake, there's only one place I need to fix.

Author: Alan Chu (ualch9@gmail.com)

```c
#include <time.h>   /* for time   */
#include <stdio.h>  /* for printf */
#include <stdlib.h> /* for rand   */

#define kNUMBER_OF_DICE 5

int random_integer(int min, int max) {
    return rand() % max + min;
}

// ... in a different file ...
void roll_dice(int* dice_arr) {
    for (int i = 0; i < kNUMBER_OF_DICE; i++) {
        dice_arr[i] = random_integer(1, 6);
    }
}

// ... in a different file ...
void print_face_values(int* dice_arr) {
    for (int i = 0; i < kNUMBER_OF_DICE; i++) {
        printf("Dice %d: %d\n", i + 1, dice_arr[i]);
    }
}

int main() {
    srand((unsigned int)time(NULL));
    int dice[kNUMBER_OF_DICE];

    roll_dice(dice);
    print_face_values(dice);

    return 0;
}
```

Woah, im separating functions into different files to keep my main function from bloating up!!!

Author: Alan Chu (ualch9@gmail.com)

BEFORE

```c
int main(void) {
    // Roll dice
    int die1 = roll_die();
    int die2 = roll_die();
    int die3 = roll_die();
    int die4 = roll_die();
    int die5 = roll_die();

    // Print dice face value
    printf("Dice 1: %d\n", die1);
    printf("Dice 2: %d\n", die2);
    printf("Dice 3: %d\n", die3);
    printf("Dice 4: %d\n", die4);
    printf("Dice 5: %d\n", die5);

    return 0;
}
```

AFTER

```c
int main() {
    int dice[kNUMBER_OF_DICE];

    roll_dice(dice);
    print_face_values(dice);

    return 0;
}
```

This code snippet only accounts for a small part of the whole program. Keeping things concise is necessary at scale.

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations