Intro to C

Allison Binger

What do you think of C?

Any takeaways from your first sprint?

What challenges did you encounter and what did you do to solve them?

Why are we learning C?

Foundation for high-level languages, like C++, Java, C#, and Python!

More interaction with hardware, manipulation of memory, and high-level concepts like data types and functions.

C is the most popular programming language in the world!

C vs Bash

C

Procedural programming

Direct access to memory for management

Syntax is more complex, but well organized (imo)

VS

Bash

Command-line interpreter

No direct control over memory allocation

Simpler syntax, more focused on command-line operations

When would we use C v.s. Bash?

C:

- Systems programming
- Operating systems
- Performance-critical applications

Bash:

- NEVER
- jk lol, I use a bash script daily!
- Scripting and automation tasks in Unix-like environments
- System commands and file manipulations

Libraries

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

```c
#include <stdlib.h>
```

```c
#include "main.h"
```

```c
int _putchar(char);
void print_alphabet(void);
void print_alphabet_x10(void);
int _islower(int c);
int _isalpha(int c);
int print_sign(int n);
int _abs(int);
int print_last_digit(int);
void jack_bauer(void);
void times_table(void);
int add(int, int);
void print_to_98(int n);
int _isupper(int c);
int _isdigit(int c);
int mul(int a, int b);
void print_numbers(void);
void print_most_numbers(void);
void more_numbers(void);
void print_line(int n);
void print_diagonal(int n);
void print_square(int size);
void print_triangle(int size);
```

My lovely friend, Betty

Named after Betty Holberton

Betty Style Documentation

```c
#include "main.h"
/**
 * _islower - checks if value c is lowercase
 * @c: character to input into stdinp
 *
 * Return: Always 0
 */
int _islower(int c)
```

Example 1:
‘if’ statements in C

HEADER FILE INCLUSION

BETTY STYLE DOCUMENTATION

FUNCTION DECLARATION

MAIN BODY

RETURN STATEMENT

```c
#include <stdio.h>
/**
 * main- entry point
 *
 * Return: void (always success)
*/
int main(void)
/* Function declaration ^ */
{
        int num = 5;
        /* variable declaration*/

        if (num < 10)
        /* conditional statement */
        {
                printf("The number %d is less than 10\n", num);
                /* print using the %d format specifier */
        }
        return (0);
        /* return statement using int (0) to indicate success */
}
```

Example 2: ‘if, else, and else if’ statements in C

HEADER FILE INCLUSION

BETTY STYLE DOCUMENTATION

FUNCTION DECLARATION

MAIN BODY

RETURN STATEMENT

```c
#include <stdio.h>
/**
 * main- entry point
 *
 * Return: void (always success)
*/
int main(void)
{
        int num = 5;
        /* variable declaration */

        if (num < 3)
        /* conditional statement */
        {
                printf("The number %d is less than 3\n", num);
                /* print using %d format specifier */
        }
        else if (num > 6)
        {
                printf("The number %d is greater than 6\n", num);
                /* print using %d format specifier */
        }
        else
        {
                printf("The number %d is between 3 and 6\n", num);
                /* print using %d format specifier */
        }
        return (0);
}
```

Example 3: ‘if, else, and else if’ statements inside of a for loop in C

HEADER FILE INCLUSION

BETTY STYLE DOCUMENTATION

FUNCTION DECLARATION

MAIN BODY

RETURN STATEMENT

```c
#include <stdio.h>
#include <unistd.h>
/**
 * main- entry point
 *
 * Return: 0 (success)
*/
int main(void)
{
        int num;
        /* declaring int variable with no value */

        for (num = 0; num < 11; num++)
        /* iteration, assigns value to num, limit for loop to stop, and increment amount */
        {
                if (num < 3)
                {
                        printf("The number %d is less than 3\n", num);
                }
                else if (num > 6)
                {
                        printf("The number %d is greater than 6\n", num);
                }
                else
                {
                        printf("The number %d is between 3 and 6\n", num);
                }
                sleep(1);
                /* waits 1 second for loop to continue */
        }
        return (0);
        /* returns int value of 0, successful */
}
```

Example 4:
while loops in C

HEADER FILE INCLUSION

BETTY STYLE DOCUMENTATION

FUNCTION DECLARATION

MAIN BODY

RETURN STATEMENT

```c
#include <stdio.h>
#include <unistd.h>
/**
 * main- entry point
 *
 * Return: 0 (success)
*/
int main(void)
{
        int num = 0;
        int limit = 10;
        /* delcares and assigns values to variables num and limit */

        while (num < limit)
        /* complete this loop until the condition is met */
        {
                printf("The number is %d, which is less than the limit: %d\n", num, limit); /* prints */
                num++; /* increments num */
                sleep(1); /* waits one second for loop */
        }
        printf("You have reached the limit of %d!\n\n", limit); /* prints once outside of loop */
        return (0); /* returns int value to indiciate success */
}
```

Example 5:
Music!

Part 1: Helper Functions

HEADER FILE INCLUSION

BETTY STYLE DOCUMENTATION

FUNCTION DECLARATION

MAIN BODY

```c
#include <stdio.h>
#include <unistd.h>
/**
 * play_maracas- gives cue for instrument
 *
 * Return: void (success)
*/
void play_maracas(void)
{
		printf("Maracas shake! ");
}

/**
 * play_tambourine- gives cue for instrument
 *
 * Return: void (success)
*/
void play_tambourine(void)
{
		printf("Tambourine beat!\n");
}
```

Part 2: Music Loop

FUNCTION DECLARATION

COUNTDOWN LOOP

MAIN MUSIC LOOP STARTS

START INTERNAL BEAT LOOP (4/4)

CONDITIONAL STATEMENTS PICK INSTRUMENTS

TRIANGLE ENDS THE SONG

RETURN STATEMENT

```c
int main(void)
{
        int beat;
        int measure = 0;
        int countdown;

        for (countdown = 5; countdown <= 8; countdown++)
        {
                /* starts loop to give the 5, 6, 7, 8 (tempo) */
                printf("%d\n", countdown);
                sleep(1);
        }
        measure++; /* moves to measure 1 */
        while (measure <= 4) /* how long is your song? this one is 4 measures long! */
        {
                printf("Measure: %d\n", measure); /* prints your measure before the beat starts */
                for (beat = 1; beat <= 4; beat++)
                /* starts your beat! this is the 4/4 time signature */
                {
                        printf("%d - ", beat); /* prints which beat you're on inside the measure */
                        if ((beat % 2) != 0) /* if the beat is odd, call the play_tambourine function */
                        {
                                play_tambourine();
                                sleep(1);
                        }
                        else /* i.e. if the beat is NOT odd (if it is even) */
                        {
                                if (beat == 4) /* check if the beat is beat #4, if it is, do this first! */
                                {
                                        play_maracas(); /* calls play_maracas function */
                                }
                                clap_along(); /* after checking for beat #4, it will carry out clap_along */
                                sleep(1);
                        }
                }
                measure++; /* increment the mesure to restart the while loop, now on next measure! */
        }
        play_triangle(); /* end the song with a triangle ding! */
        return (0); /* return 0 to indicate successful loop */
}
```