C: output; control structures

COSC 208, Introduction to Computer Systems, 2021-09-01

Announcements

- Pre-class questions: doing well, couple needs to retry (not falling means not improving, *ski anology*)
- Lab 1: today & tomorrow

Outline

- Warm-up
- Output
- Control structures

Warm-up

Q1: *Write a program that prints the number of days, hours, and minutes in a week.*

```
#include <stdio.h>
int main() {
    int days = 7;
    int hours = days * 24;
    int minutes = hours * 60;
    printf("1 week = %d days = %d hours = %d minutes\n", days, hours,
minutes);
}
```

Output

- What is the syntax for `printf`?
  - `printf(FORMAT_STRING, VALUES, ...);`
  - `FORMAT_STRING` is a string constant (sequence of characters surrounded by double quotes) that may optionally include format specifiers
  - Format specifiers define how to convert a value to a string
    - `%d` decimal (i.e., base 10) number
    - `%c` character
    - `%x` hexadecimal (i.e., base 16) number
    - `%s` string — more on this next week
  - After the format string, include a value for each format specifier
    - A compile error will occur if the number of format specifiers does not match the number of values
    - A compile warning will occur if the value type does not match the format specifier
- Next week we'll talk about reading input using `fgets`
  - *Dive into Systems* uses `scanf`, which has several disadvantages

Output practice

- Q2: *Use `printf` to output today's date (e.g., `2021-1-29`)*

```
printf("%d-%d-%d\n", 2021, 1, 29);
```

- Q3: *Use `printf` to output the dimensions of a college basketball court (`94ft x 50ft`)*

```
printf("%dft x %dft\n", 94, 50);
```

- Q4: *Use `printf` to output a patient's first and last initial and systolic and diastolic blood pressure (e.g., `A.G. 115/70`)*

```
printf("%c.%c. %d/%d\n", first, last, systolic, diastolic);
```

Control structures

- Control structures in C have the same syntax as control structures in Java.
    - Conditionals

```
if (/* BOOLEAN EXPRESSION */) { // Exactly one
    /* STATEMENTS */
}
else if (/* BOOLEAN EXPRESSION */) { // Zero or more
    /* STATEMENTS */
}
else { // Zero or one
    /* STATEMENTS */
}
```

    - For loops

```
for (/* INTITIALIZER */; /* CONTINUATION CONDITION */; /* UPDATE
*/) {
    /* STATEMENTS */
}
```

- While loops

```
while (/* CONTINUATION CONDITION */) {
    /* STATEMENTS */
}
```

- Curly braces are optional if the body of a conditional, for loop, or while loop is only one line
  - But, you should **always** include them to make the code easier to read and reduce the likelihood of future errors

Practice

- Q5: *Write a program that flips a coin: call `random()` to generate a random number, and print `heads` if the number is even and `tails` if the number is odd.*

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int num = random();
    if (num % 2 == 0) {
        printf("heads\n");
    } else {
        printf("tails\n");
    }
}
```

- Q6: *Write a program that prints all even numbers from 1 to 100 using a for loop.*

```
#include <stdio.h>
int main() {
    for (int i = 2; i <= 100; i+=2) {
        printf("%d\n", i);
    }
}
```

~ OR ~

```
int main() {
    for (int i = 0; i <= 100; i++) {
        if (0 == i % 2) {
            printf("%d\n", i);
        }
    }
}
```

- Q7: *Write a program that prints every letter of the alphabet in upper and lower case:* `AaBbCcDd...YyZz`

```c
#include <stdio.h>
int main() {
    for (char upper = 'A'; upper <= 'Z'; upper++) {
        char lower = upper - 'A' + 'a';
        printf("%c%c", upper, lower);
    }
    printf("\n");
}
```

- Q8: *Write a program that prints all numbers from 1 to 100, except:*
    - *If the number is divisible by 3 then print* `Three`
    - *If the number is divisible by 5 then print* `Five`
    - *If the number is divisible by 3 and 5, print* `Both`

```c
#include <stdio.h>
int main() {
    for (int i = 1; i <= 100; i++) {
        if (i % 3 == 0 && i % 5 == 0) {
            printf("Both\n");
        }
        else if (i % 5 == 0) {
            printf("Five\n");
        }
        else if (i % 3 == 0) {
            printf("Three\n");
        }
        else {
            printf("%d\n", i);
        }
    }
}
```

Extra practice

- Q9: *Write a program that prints out the powers of 2 from 2 through 2048.*

```c
#include <stdio.h>
#define MAX 2048
int main() {
    int i = 2;
    while (i <= MAX) {
        printf("%d ", i);
        i *= 2;
    }
    printf("\n");
}
```

- Q10: *Write the same program using a while loop.*

```c
#include <stdio.h>
int main() {
    int i = 2;
    while (i <= 100) {
        printf("%d\n", i);
        i+=2;
    }
}
```