Strings

Darin Brezeale

The University of Texas at Arlington

The `char` Data Type

A `char` is a one byte integer type typically used for storing characters.

Example:

```
char oneLetter = 'D';
```

We enclose the character in single quotes, not double quotes.

ASCII

Plain text is often stored in a format called ASCII (American Standard Code for Information Interchange). There are 128 characters in ASCII, which includes the standard alphanumeric characters as well as some non-printable characters (e.g., tab, newline, and so forth).

char Example

ASCII characters can be stored using the character or the decimal value of the character.

```
#include <stdio.h>

int main(void)
{
   char letter_a = 'a';  /* note the use of single quotes */
   char decimal_a = 97;  /* 97 is the decimal value
                            for 'a' in ASCII */

   /* note the %c to print characters */
   printf("letter_a is %d and %c\n", letter_a, letter_a);
   printf("decimal_a is %d and %c\n", decimal_a, decimal_a);
}
```

Output

```
letter_a is 97 and a
decimal_a is 97 and a
```

char cont.

If we check an ASCII chart, we see that the letters A–Z have ASCII values of 65–90 while a–z have values of 97–122. Because the letters are stored as numbers, we can perform numeric operations on them.

```
#include <stdio.h>

int main(void)
{
   char uppercase = 'A';  /* 65 in ASCII */
   char lowercase;

   lowercase = uppercase + 32;
   printf("Adding 32 to %c gives us %c\n", uppercase, lowercase);
}
```

prints

```
Adding 32 to A gives us a
```

Array of `char`

C does not have a string type. Instead, we use an array of `char`s for storing a string.
We can declare this just like we did with other types.

Example

```
char letters[10];
```

Strings

Storing each character individually is a tedious process. We can do this simultaneously by entering a string.

Example

```
char myName[] = "Darin Brezeale";
```

This time we DO use double quotes.

Strings cont.

There is a difference between the two approaches. When we create a string, a null character (i.e., $\backslash 0$) is added to the end. Note that when writing the null character, you should use a zero, not the letter O.

A null allows the compiler to know where the string ends so that we can print it by name.

Example

```
char myName[] = "Darin Brezeale";

printf("My name is %s", myName);  /* print using %s */
```

will print

```
Darin Brezeale
```