

MC140: lecture #11

today's topic:

character review
strings
string library

2/14/01 12:54 PM

1

characters: review.

- storage
 - as 1 byte = 8 bits
 - numeric values from 0 to 127, used as indexes into the ASCII table
- representation
 - character constants are single characters, surrounded by single quotes
- data type: **char**
- format sequence: %c

2/14/01 12:54 PM

2

characters: interpretation.

- a numeric value is stored, and this value is *interpreted* as a character
- example:

```
int x = 97;
char y = 'a';
printf( "x = %d %c\n", x, x );
printf( "y = %d %c\n", y, y );
```
- output:

```
97 a
97 a
```

2/14/01 12:54 PM

3

printing a row of characters.

```
#include <stdio.h>
int main( void ) {
    int i;
    char y = '*';
    for ( i=1; i<=10; i++ ) {
        printf( "%c", y );
    } /* end of for */
    printf( "\n" );
} /* end of main() */
```

2/14/01 12:54 PM

4

using a function to print a row of chars.

```
declaration — #include <stdio.h>
               void printRowOfChars( void );
               int main( void ) {
call           printRowOfChars();
               } /* end of main() */
definition — void printRowOfChars( void ) {
               int i;
               char y = '*';
               for ( i=1; i<=10; i++ ) {
                   printf( "%c", y );
               } /* end of for */
               printf( "\n" );
               } /* end of printRowOfChars() */
```

2/14/01 12:54 PM

5

printing 2 rows of characters.

```
#include <stdio.h>
void printRowOfChars( void );
int main( void ) {
    int j;
    for ( j=1; j<=2; j++ ) {
        printRowOfChars();
    } /* end for */
} /* end of main() */
void printRowOfChars( void ) {
    int i;
    char y = '*';
    for ( i=1; i<=10; i++ ) {
        printf( "%c", y );
    } /* end of for */
    printf( "\n" );
} /* end of printRowOfChars() */
```

2/14/01 12:54 PM

6

strings.

- storing multiple characters in a single variable
- data type is still char
- BUT it has a *length*
- last character is *terminator* : `'\0'`
- string constants are surrounded by *double* quotes
- example:
`char s[6] = "ABCDE";`

2/14/01 12:54 PM

7

strings, 2.

- example:
`char s[6] = "ABCDE";`
- storage looks like this:

A	B	C	D	E	\0
---	---	---	---	---	----

2/14/01 12:54 PM

8

strings: printing.

- format sequence: `%s`
- example:

```
#include <stdio.h>
int main( void ) {
    char str[6] = "ABCDE";
    printf( "str = %s\n", str );
} /* end of main() */
```
- output:
`ABCDE`

2/14/01 12:54 PM

9

character functions.

- character handling library
`#include <ctype.h>`
- functions include:
`int isdigit( int c );`
`int isalpha( int c );`
`int isspace( int c );`
`int tolower( int c );`
`int toupper( int c );`
- DD: p 309-310

2/14/01 12:54 PM

10

string functions.

- string handling library
`#include <string.h>`
- functions include:
`int strlen( char *s );`
- many more functions, which we'll cover another day...

2/14/01 12:54 PM

11

reading.

- material covered today:
 - DD: 8.1-8.3
- *assignment #4 is due Monday 26 Feb*

2/14/01 12:54 PM

12