CCTYPE Library

Math and Computer Science

SOUTH DAKOTA
SCHOOL OF MINES
& TECHNOLOGY

What is it

- c in cctype means it is from the C language.
- Contains many functions that preform operations on characters
- To use the library:

```
#include <cctype>
```

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Conversion Functions

- tolower – Converts an uppercase character (A-Z) to a lower case characters. Any other character is left unchanged. The result is returned.

```
int tolower (int c);

char ch = 'R';
char result;
result = tolower(ch);  // ch is typecasted implicitly to integer.
                       // result is now the character 'r'
                       // ch is still 'R'
```

Conversion Functions

- toupper – Converts an lowercase character (a-z) to a uppercase characters. Any other character is left unchanged. The result is returned.

```
int toupper (int c);

char ch = 'r';
char result;
result = toupper(ch);  // ch is typecasted implicitly to integer.
                       // result is now the character 'R'
                       // ch is still 'r'
```

Classification Functions

- isalnum – alphanumeric classification
  - Digits '0' – '9'
  - Uppercase 'A' – 'Z'
  - Lowercase 'a' – 'z'

```
int isalnum( int c );

char ch = '9';
int result;
result = isalnum( ch ) ;  // char is implicitly type casted to an integer
                          // result is non zero for true, 0 for false
                          // in this example, result is non zero
```

Classification Functions

- isalpha – alphabetic classification
  - Uppercase 'A' – 'Z'
  - Lowercase 'a' – 'z'

```
int isalpha( int c );

char ch = 'M';
int result;
result = isalpha( ch ) ;   // char is implicitly type casted to an integer
                           // result is non zero for true, 0 for false
                           // in this example, result is non zero
```

Classification Functions

- isblank – blank classification
  - Tab '\t'
  - Space ' ' ← space in between single quotes

```
int isblank( int c );

char ch = '  '; ← space in between single quotes
int result;
result = isblank( ch ) ;   // char is implicitly type casted to an integer
                           // result is non zero for true, 0 for false
                           // in this example, result is non zero
```

Classification Functions

- isdigit – numeric classification
  - Digits ('0' – '9')

```
int isdigit( int c );

char ch = 'a';
int result;
result = isdigit( ch ) ;     // char is implicitly type casted to an integer
                             // result is non zero for true, 0 for false
                             // in this example, result is zero
```

Classification Functions

- islower – lower case classification
    - Characters ('a' – 'z')

```
int islower( int c );

char ch = 'd';
int result;
result = islower( ch ) ;   // char is implicitly type casted to an integer
                           // result is non zero for true, 0 for false
                           // in this example, result is a non zero
```

Classification Functions

- isupper – Upper case classification
  - Characters ('A' – 'Z')

```
int isupper( int c );

char ch = 'Y';
int result;
result = isupper( ch ) ;   // char is implicitly type casted to an integer
                           // result is non zero for true, 0 for false
                           // in this example, result is a non zero
```

Classification Functions

- ispunct – punctuation character
  - !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~

```
int ispunct ( int c );

char ch = '?';
int result;
result = ispunct( ch ) ;   // char is implicitly type casted to an integer
                           // result is non zero for true, 0 for false
                           // in this example, result is a non zero
```

Classification Functions

- isspace – white space classification
  - Tab '\t'
  - Space ' ' ← space in between single quotes
  - Control '\f', '\v', '\n', '\r'

```
int isspace( int c );

char ch = ' '; ← space in between single quotes
int result;
result = isspace( ch ) ;   // char is implicitly type casted to an integer
                             // result is non zero for true, 0 for false
                             // in this example, result is non zero
```

-

Classification Functions

- iscntrl – Control character.
  - Values (0 – 8, 14 - 31)
  - Tab '\t'
  - Whitespace control ('\f', '\v', '\n', '\r')

```
int iscntrl( int c );

char ch = '\t';
int result;
result = iscntrl( ch ) ;     // char is implicitly type casted to an integer
                             // result is non zero for true, 0 for false
                             // in this example, result is non zero
```

SOUTH DAKOTA
M
SCHOOL OF MINES
& TECHNOLOGY

Classification Functions

- isgraph – graphical representation
  - !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
  - Digits ('0' – '9')
  - Characters ('A'-'Z' & 'a' – 'z')

```
int isgraph( int c );

char ch = '#';
int result;
result = isgraph( ch ) ;     // char is implicitly type casted to an integer
                             // result is non zero for true, 0 for false
                             // in this example, result is a non zero
```

Classification Functions

- isprint – Is the character printable it is the opposite of iscntrl
  - Space (' ') ← space character between single quotes
  - Digits ('0' – '9')
  - Alphabet ('A' – 'Z', 'a' – 'z')
  - !"#$%&'()*+,-./ :;<=>?@ [\]^_` {|}~

```
int isprint( int c );

char ch = '$';
int result;
result = isprint( ch ) ;      // char is implicitly type casted to an integer
                              // result is non zero for true, 0 for false
                              // in this example, result is a non zero
```

Classification Functions

- isxdigit – Characters used in representing hexadecimal numbers
  - Digits ('0' – '9')
  - Characters ('A'-'F', 'a' – 'f')

```
int isxdigit( int c );

char ch = 'c';
int result;
result = isxdigit( ch );      // char is implicitly type casted to an integer
                              // result is non zero for true, 0 for false
                              // in this example, result is a non zero
```

Notes

- Remember these functions
  - toupper, tolower
  - isupper, islower, isalnum, isalpha isspace, ispunct
- Be aware of what exists out in the cctype library.
- Do not rewrite the functions that exist without good reason.