**Chapter 10**

**Character Testing, String Library Functions, String Processing**

**Overview**

There are several categories for string processing:

* + Character testing
  + Character conversion
  + String Library

Example:

//Character testing function

string str = "7-Eleven";

//Must include cctype to use these macros

//isalpha

if ( ((str[0] >= 'A') && (str[0] <= 'Z')) || ((str[0] >= 'a') && (str[0] <= 'z')) )

{

return((int)str[0])

}

else

{

return 0;

}

//isalnum

if ( ((str[0] >= 'A') && (str[0] <= 'Z')) || ((str[0] >= 'a') && (str[0] <= 'z')) ||

((str[0] >= '0') && (str[0] <= '9')) )

{

cout << "Alpha/num found" << endl;

}

//All others are done in a similar way!

.

Can get the # of characters stored in string data type using

cout << str.length() << endl; //length is a special method that can be invoked on str

Can return the array of characters from the string data type using

const char\* str1 = str.c\_str();

W/hat does const means in this context?

You can assign one string to another using

string str = "7-Eleven";

string copy;

copy = str;

//Can also append 1 character at a time using the following

for (i = 0; i < (int)str.length(); i++)

{

copy += str[i]; //This throws an exception

}

//Can also append strings using +

string new\_string = str + upper;

**Character Testing**

require cctype header file

|  |  |
| --- | --- |
| FUNCTION | MEANING |
| isalpha | true if arg. is a letter, false otherwise |
| isalnum | true if arg. is a letter or digit, false otherwise |
| isdigit | true if arg. is a digit 0-9, false otherwise |
| islower | true if arg. is lowercase letter, false otherwise |
| isprint | true if arg. is a printable character, false otherwise |
| ispunct | true if arg. is a punctuation character, false otherwise |
| isupper | true if arg. is an uppercase letter, false otherwise |
| isspace | true if arg. is a whitespace character, false otherwise |

**Character Conversion**

* Require cctype header file
* Functions:

toupper: if char argument is lowercase letter, return uppercase equivalent; otherwise,

return input unchanged

char ch1 = 'H';  
char ch2 = 'e';  
char ch3 = '!';

cout << toupper(ch1); // displays 'H'

cout << toupper(ch2); // displays 'E'

cout << toupper(ch3); // displays '!'

tolower: if char argument is uppercase letter, return lowercase equivalent; otherwise,

return input unchanged

char ch1 = 'H';  
char ch2 = 'e';  
char ch3 = '!';

cout << tolower(ch1); // displays 'h'

cout << tolower(ch2); // displays 'e'

cout << tolower(ch3); // displays '!'

//Can use with [] operator of string

for (i = 0; i < (int)str.length(); i++)

{

str[i]= toupper(str[i]);

}

**Library Functions**

* Require the string header file
* Functions take one or more strings as arguments. Can use:
  + string name

Functions:

length() - returns length of string

Example:

string city = "Missoula";

cout << city.length() << endl;// prints 8

appends str2 to the end of str1

string location = "Missoula, ";

string state = "MT";

location += state

copies str2 to str1  
  
string name = “Joe”;

string fname;

fname = name;

:

**String/Numeric Conversion Functions**

require cstdlib header file

|  |  |  |
| --- | --- | --- |
| FUNCTION | PARAMETER | ACTION |
| atoi | C-string | converts C-string to an int value, returns the value |
| atol | C-string | converts C-string to a long value, returns the value |
| atof | C-string | converts C-string to a double value, returns the value |
| Itoa | int,C-string, int | converts 1st int parameter to a C-string, stores it in 2nd parameter. 3rd parameter is base of converted value |
| stoi | string | Converts a string to an int |
| stod | string | Converts a string to a float or a double |

**Using getline to read all data**

The suggested policy for reading input data is:

If getline(cin,string\_var) is used once anywhere in a program, use it for reading all data!

Do not use cin >> for reading numeric data – use getline(cin, string\_var)

Do not mix cin >> and getline in the same program.

We will use the following functions to convert a string to numeric:

int val\_int = stoi(string\_var);

double val\_double = stod(string\_var);

Where string\_var contains a string that represents a number.

Will need to perform the following steps:

1. Read all numeric data using getline

//Prompt for a double first

cout << "Enter a double: ";

getline(cin,data);

1. Convert the string that represents a number to either a double or int

result = stod(data);

//Prompt for an integer next

cout << "Enter an integer: ";

getline(cin,data);

result1 = stoi(data);

//Prompt for a name

cout << "Enter first name:";

getline(cin,data);

cout << data << endl;

//Prompt for last name

cout << "Enter last name:";

getline(cin, data);

cout << data << endl;

Otherwise will need to read an additional getline in between the cin>> and the first getline.

Can display string variable data using gout as shown below:

gout << setPos(200,200) << (char\*)data.c\_str() <<endg;

Can convert numerics to a string using the following to create a file name

string fn;

int card\_no = 4;

fn = string("h") + to\_string(card\_no) + string(".bmp");