Appendix I: Header File and Library Function Reference

This appendix provides a reference for the C++ library functions discussed in the book. The following table gives an alphabetical list of functions. Tables of functions that are organized by their header files follow it.

Table I-1 Alphabetical Listing of Selected Library Functions

Function	Details
`abs(m)`	***Header File:*** `cmath` ***Description:*** Accepts an integer argument. Returns the absolute value of the argument as an integer. ***Example:*** `a = abs(m);`
`atof(str)`	***Header File:*** `cstdlib` ***Description:*** Accepts a C-string as an argument. The function converts the string to a `double` and returns that value. ***Example:*** `num = atof("3.14159");`
`atoi(str)`	***Header File:*** `cstdlib` ***Description:*** Accepts a C-string as an argument. The function converts the string to an `int` and returns that value. ***Example:*** `num = atoi("4569");`
`atol(str)`	***Header File:*** `cstdlib` ***Description:*** Accepts a C-string as an argument. The function converts the string to a `long` and returns that value. ***Example:*** `num = atol("5000000");`

Table I-1 Alphabetical Listing of Selected Library Functions *(continued)*

Function	Details
`cos(m)`	***Header File:*** `cmath` ***Description:*** Accepts a `double` argument. Returns the cosine of the argument. The argument should be an angle expressed in radians. The return type is `double`. ***Example:*** `a = cos(m);`
`exit(status)`	***Header File:*** `cstdlib` ***Description:*** Accepts an `int` argument. Terminates the program and passes the value of the argument to the operating system. ***Example:*** `exit(0);`
`exp(m)`	***Header File:*** `cmath` ***Description:*** Accepts a `double` argument. Computes the exponential function of the argument, which is e^x. The return type is `double`. ***Example:*** `a = exp(m);`
`fmod(m, n)`	***Header File:*** `cmath` ***Description:*** Accepts two `double` arguments. Returns, as a `double`, the remainder of the first argument divided by the second argument. Works like the modulus operator, but the arguments are doubles. (The modulus operator only works with integers.) Take care not to pass zero as the second argument. Doing so would cause division by zero. ***Example:*** `a = fmod(m, n);`

`isalnum(ch)`	***Header File:*** `cctype`
	Description:
	Accepts a `char` argument. Returns true if the argument is a letter of the alphabet or a digit. Otherwise, it returns false.
	Example:
	`if (isalnum(ch))`
	`  cout << ch << " is alphanumeric.\n";`

`isdigit(ch)`	***Header File:*** `cctype`
	Description:
	Accepts a `char` argument. Returns true if the argument is a digit 0 - 9. Otherwise, it returns false.
	Example:
	`if (isdigit(ch))`
	`  cout << ch << " is a digit.\n";`

Table I-1 Alphabetical Listing of Selected Library Functions *(continued)*

Function	Details

`islower(ch)`	***Header File:*** `cctype`
	Description:
	Accepts a `char` argument. Returns true if the argument is a lowercase letter. Otherwise, it returns false.
	Example:
	`if (islower(ch))`
	`  cout << ch << " is lowercase.\n";`

`isprint(ch)`	***Header File:*** `cctype`
	Description:
	Accepts a `char` argument. Returns true if the argument is a printable character (including a space). Returns false otherwise.
	Example:
	`if (isprint(ch))`
	`  cout << ch << " is printable.\n";`

`ispunct(ch)`	***Header File:*** `cctype`
	Description:
	Accepts a `char` argument. Returns true if the argument is a printable character other than a digit, letter, or space. Returns false otherwise.
	Example:
	`if (ispunct(ch))`
	`  cout << ch << " is punctuation.\n";`

`isspace(ch)`	***Header File:*** `cctype`
	Description:
	Accepts a `char` argument. Returns true if the argument is a whitespace character. Whitespace characters are any of the following:
	• space..... ' '
	• newline..... '\n'
	• tab..... '\t'
	• vertical tab..... '\v'
	Otherwise, it returns false.
	Example:
	`if (isspace(ch))`
	`  cout << ch << " is whitespace.\n";`

`isupper(ch)`	***Header File:*** `cctype`
	Description:
	Accepts a `char` argument. Returns true if the argument is an uppercase letter. Otherwise, it returns false.
	Example:
	`if (isupper(ch))`
	`  cout << ch << " is uppercase.\n";`

`log(m)`	***Header File:*** `cmath` ***Description:*** Accepts a `double` argument. Returns, as a `double`, the natural logarithm of the argument. ***Example:*** `a = log(m);`

Table I-1 Alphabetical Listing of Selected Library Functions *(continued)*

Function	Details
`log10(m)`	***Header File:*** `cmath` ***Description:*** Accepts a `double` argument. Returns, as a `double`, the base-10 logarithm of the argument. ***Example:*** `a = log10(m);`
`pow(m, n)`	***Header File:*** `cmath` ***Description:*** Accepts two `double` arguments. Returns the value of argument 1 raised to the power of argument 2. ***Example:*** `a = pow(m, n);`
`rand()`	***Header File:*** `cstdlib` ***Description:*** Generates a pseudorandom number. ***Example:*** `x = rand();`
`sin(m)`	***Header File:*** `cmath` ***Description:*** Accepts a `double` argument. Returns, as a `double`, the sine of the argument. The argument should be an angle expressed in radians. ***Example:*** `a = sin(m);`
`sqrt(m)`	***Header File:*** `cmath` ***Description:*** Accepts a `double` argument. Returns, as a `double`, the square root of the argument. ***Example:*** `a = sqrt(m);`
`srand(m)`	***Header File:*** `cstdlib` ***Description:*** Accepts an `unsigned int` argument. The argument is used as a seed value to randomize the results of the `rand()` function. ***Example:*** `srand(m);`
`strcat(str1, str2)`	***Header File:*** `cstring` ***Description:*** Accepts two C-strings as arguments. The function appends the contents of the second string to the first string. (The first string is altered; the second string is left unchanged.) ***Example:*** `strcat(string1, string2);`

Table I-1 Alphabetical Listing of Selected Library Functions *(continued)*

Function	Details

`strcmp(str1, str2)`	***Header File:*** `cstring`
	Description:
	Accepts pointers to two string arguments. If string1 and string2 are the same, this function returns 0. If string2 is alphabetically greater than string1, it returns a positive number. If string2 is alphabetically less than string1, it returns a negative number.
	Example:
	`if (strcmp(string1, string2) == 0)`
	`  cout << "The strings are equal.\n";`

`strcpy(str1, str2)`	***Header File:*** `cstring` ***Description:*** Accepts two C-strings as arguments. The function copies the second string to the first string. The second string is left unchanged. ***Example:*** `strcpy(string1, string2);`
`strlen(str)`	***Header File:*** `cstring` ***Description:*** Accepts a C-string as an argument. Returns the length of the string (not including the null terminator). ***Example:*** `len = strlen(name);`
`strncpy(str1, str2, n)`	***Header File:*** `cstring` ***Description:*** Accepts two C-strings and an integer argument. The third argument, an integer, indicates how many characters to copy from the second string to the first string. If string2 has fewer than n characters, string1 is padded with '\0' characters. ***Example:*** `strncpy(string1, string2, n);`
`strstr(str1, str2)`	***Header File:*** `cstring` ***Description:*** Searches for the first occurrence of string2 in string1. If an occurrence of string2 is found, the function returns a pointer to it. Otherwise, it returns a NULL pointer (address 0). ***Example:*** `cout << strstr(string1, string2);`
`tan(m)`	***Header File:*** `cmath` ***Description:*** Accepts a `double` argument. Returns, as a double, the tangent of the argument. The argument should be an angle expressed in radians. ***Example:*** `a = tan(m);`

Table I-1 Alphabetical Listing of Selected Library Functions *(continued)*

Function	Details
`tolower(ch)`	***Header File:*** `cctype` ***Description:*** Accepts a `char` argument. Returns the lowercase equivalent of its argument. ***Example:*** `ch = tolower(ch);`
`toupper(ch)`	***Header File:*** `cctype` ***Description:*** Accepts a `char` argument. Returns the uppercase equivalent of its argument. ***Example:*** `ch = toupper(ch);`

Table I-2 Selected `cstdlib` functions

Function	Details
`atof(str)`	***Header File:*** `cstdlib` ***Description:*** Accepts a C-string as an argument. The function converts the string to a `double` and returns that value. ***Example:*** `num = atof("3.14159");`
`atoi(str)`	***Header File:*** `cstdlib` ***Description:*** Accepts a C-string as an argument. The function converts the string to an `int` and returns that value. ***Example:*** `num = atoi("4569");`
`atol(str)`	***Header File:*** `cstdlib` ***Description:*** Accepts a C-string as an argument. The function converts the string to a `long` and returns that value. ***Example:*** `num = atol("5000000");`
`exit(status)`	***Header File:*** `cstdlib` ***Description:*** Accepts an `int` argument. Terminates the program and passes the value of the argument to the operating system. ***Example:*** `exit(0);`

Table I-2 Selected `cstdlib` functions *(continued)*

Function	Details
`rand()`	***Header File:*** `cstdlib` ***Description:*** Generates a pseudorandom number. ***Example:*** `x = rand();`
`srand(m)`	***Header File:*** `cstdlib` ***Description:*** Accepts an `unsigned int` argument. The argument is used as a seed value to randomize the results of the `rand()` function. ***Example:*** `srand(m);`

Table I-3 Selected `cmath` Functions

Function	Details
`abs(m)`	***Header File:*** `cmath` ***Description:*** Accepts an integer argument. Returns the absolute value of the argument as an integer. ***Example:*** `a = abs(m);`
`cos(m)`	***Header File:*** `cmath` ***Description:*** Accepts a `double` argument. Returns the cosine of the argument. The argument should be an angle expressed in radians. The return type is `double`. ***Example:*** `a = cos(m);`
`exp(m)`	***Header File:*** `cmath` ***Description:*** Accepts a `double` argument. Computes the exponential function of the argument, which is e^x. The return type is `double`. ***Example:*** `a = exp(m);`
`fmod(m, n)`	***Header File:*** `cmath` ***Description:*** Accepts two `double` arguments. Returns, as a `double`, the remainder of the first argument divided by the second argument. Works like the modulus operator, but the arguments are doubles. (The modulus operator only works with integers.) Take care not to pass zero as the second argument. Doing so would cause division by zero. ***Example:*** `a = fmod(m, n);`

Table I-3 Selected `cmath` Functions *(continued)*

Function	Details
`log(m)`	***Header File:*** `cmath` ***Description:*** Accepts a `double` argument. Returns, as a `double`, the natural logarithm of the argument. ***Example:*** `a = log(m);`
`log10(m)`	***Header File:*** `cmath` ***Description:*** Accepts a `double` argument. Returns, as a `double`, the base-10 logarithm of the argument. ***Example:*** `a = log10(m);`
`pow(m, n)`	***Header File:*** `cmath` ***Description:*** Accepts two `double` arguments. Returns the value of argument 1 raised to the power of argument 2. ***Example:*** `a = pow(m, n);`
`sin(m)`	***Header File:*** `cmath` ***Description:*** Accepts a `double` argument. Returns, as a `double`, the sine of the argument. The argument should be an angle expressed in radians. ***Example:*** `a = sin(m);`
`sqrt(m)`	***Header File:*** `cmath` ***Description:*** Accepts a `double` argument. Returns, as a `double`, the square root of the argument. ***Example:*** `a = sqrt(m);`
`tan(m)`	***Header File:*** `cmath` ***Description:*** Accepts a `double` argument. Returns, as a double, the tangent of the argument. The argument should be an angle expressed in radians. ***Example:*** `a = tan(m);`

Table I-4 Selected `cstring` Functions

Function	Details
`strcat(str1, str2)`	***Header File:*** `cstring` ***Description:*** Accepts two C-strings as arguments. The function appends the contents of the second string to the first string. (The first string is altered; the second string is left unchanged.) ***Example:*** `strcat(string1, string2);`

`strcmp(str1, str2)`	***Header File:*** `cstring`
	Description:
	Accepts pointers to two string arguments. If string1 and string2 are the same, this function returns 0. If string2 is alphabetically greater than string1, it returns a positive number. If string2 is alphabetically less than string1, it returns a negative number.
	Example:
	`if (strcmp(string1, string2) == 0)`
	`  cout << "The strings are equal.\n";`

`strcpy(str1, str2)`	***Header File:*** `cstring` ***Description:*** Accepts two C-strings as arguments. The function copies the second string to the first string. The second string is left unchanged. ***Example:*** `strcpy(string1, string2);`
`strlen(str)`	***Header File:*** `cstring` ***Description:*** Accepts a C-string as an argument. Returns the length of the string (not including the null terminator) ***Example:*** `len = strlen(name);`
`strncpy(str1, str2, n)`	***Header File:*** `cstring` ***Description:*** Accepts two C-strings and an integer argument. The third argument, an integer, indicates how many characters to copy from the second string to the first string. If string2 has fewer than n characters, string1 is padded with '\0' characters. ***Example:*** `strncpy(string1, string2, n);`
`strstr(str1, str2)`	***Header File:*** `cstring` ***Description:*** Searches for the first occurrence of string2 in string1. If an occurrence of string2 is found, the function returns a pointer to it. Otherwise, it returns a NULL pointer (address 0). ***Example:*** `cout << strstr(string1, string2);`

Table I-5 Selected `cctype` Functions

Function	Details

`isalnum(ch)`	***Header File:*** `cctype`
	Description:
	Accepts a `char` argument. Returns true if the argument is a letter of the alphabet or a digit. Otherwise, it returns false.
	Example:
	`if (isalnum(ch))`
	`  cout << ch << " is alphanumeric.\n";`

`isdigit(ch)`	***Header File:*** `cctype`
	Description:
	Accepts a `char` argument. Returns true if the argument is a digit 0 - 9. Otherwise, it returns false.
	Example:
	`if (isdigit(ch))`
	`  cout << ch << " is a digit.\n";`

`islower(ch)`	***Header File:*** `cctype`
	Description:
	Accepts a `char` argument. Returns true if the argument is a lowercase letter. Otherwise, it returns false.
	Example:
	`if (islower(ch))`
	`  cout << ch << " is lowercase.\n";`

`isprint(ch)`	***Header File:*** `cctype`
	Description:
	Accepts a `char` argument. Returns true if the argument is a printable character (including a space). Returns false otherwise.
	Example:
	`if (isprint(ch))`
	`  cout << ch << " is printable.\n";`

`ispunct(ch)`	***Header File:*** `cctype`
	Description:
	Accepts a `char` argument. Returns true if the argument is a printable character other than a digit, letter, or space. Returns false otherwise.
	Example:
	`if (ispunct(ch))`
	`  cout << ch << " is punctuation.\n";`

`isspace(ch)`	***Header File:*** `cctype`
	Description:
	Accepts a `char` argument. Returns true if the argument is a whitespace character. Whitespace characters are any of the following:
	• space..... ' '
	• newline..... '\n'
	• tab..... '\t'
	• vertical tab..... '\v'
	Otherwise, it returns false.
	Example:
	`if (isspace(ch))`
	`  cout << ch << " is whitespace.\n";`

Table I-5 Selected `cctype` Functions *(continued)*

Function	Details

`isupper(ch)`	***Header File:*** `cctype`
	Description:
	Accepts a `char` argument. Returns true if the argument is an uppercase letter. Otherwise, it returns false.
	Example:
	`if (isupper(ch))`
	`  cout << ch << " is uppercase.\n";`

`tolower(ch)`	***Header File:*** `cctype` ***Description:*** Accepts a `char` argument. Returns the lowercase equivalent of its argument. ***Example:*** `ch = tolower(ch);`
`toupper(ch)`	***Header File:*** `cctype` ***Description:*** Accepts a `char` argument. Returns the uppercase equivalent of its argument. ***Example:*** `ch = toupper(ch);`