



Character Testing

• Requires 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

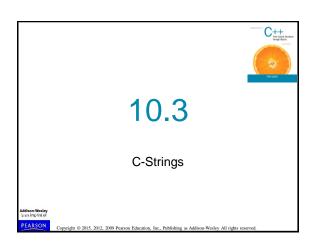
From Program 10-1 10 cout << "Enter any character: "; cin.get(input); 12 cout << "The character you entered is: " << input << endl; 13 if (isalpha(input)) cout << "That's an alphabetic character.\n"; 15 if (isdigit(input)) cout << "That's a numeric digit.\n"; 17 if (islower(input)) cout << "The letter you entered is lowercase.\n"; if (isuper(input)) cout << "The letter you entered is uppercase.\n"; if (isupec(input)) cout << "The letter you entered is uppercase.\n"; if (isupace(input)) cout << "That's a whitespace character.\n"; Addition.Weslay isologists.



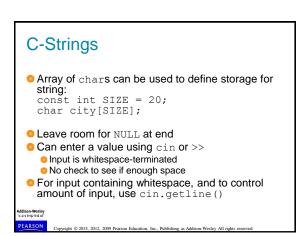
Character Case Conversion Require cctype header file Functions: toupper: if char argument is lowercase letter, return uppercase equivalent; otherwise, return input unchanged char ch1 = 'H'; char ch2 = 'e'; char ch3 = '!'; cout << toupper(ch1); // displays 'H' cout << toupper(ch2); // displays 'E' serimpted Copyright 0 2015, 2012, 2009 Pearum Education, Inc., Publishing as Addition-Wesley All rights reserved

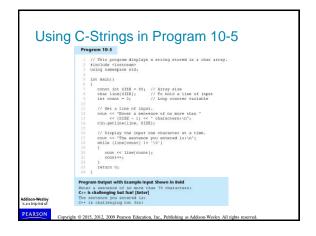
Character Case Conversion • Functions: tolower: if char argument is uppercase letter, return lowercase equivalent; otherwise, return input unchanged char ch1 = 'H'; char ch2 = 'e'; char ch3 = '!'; cout << tolower(ch1); // displays 'h' cout << tolower(ch2); // displays 'e' cout << tolower(ch3); // displays '!'

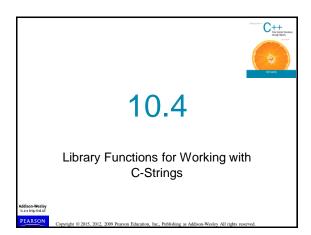
Copyright © 2015, 2012, 2009 Pearson Education, Inc., Publishing as Addison-Wesley All rights



C-String: sequence of characters stored in adjacent memory locations and terminated by NULL character String literal (string constant): sequence of characters enclosed in double quotes " ": "Hi there!" Hi i there!" Copyright © 2015, 2012, 2009 Pearson Education, Inc., Publishing an Addison-Wesley All rights nearroot.







Library Functions for Working with C-Strings

- Require the cstring header file
- Functions take one or more C-strings as arguments. Can use:
 - C-string name
 - pointer to C-string
 - literal string

Addison-Wesler is an imprint of

PEARSON

Copyright © 2015, 2012, 2009 Pearson Education, Inc., Publishing as Addison-Wesley All rights reserved.

Library Functions for Working with C-Strings

Functions:

- ostrlen(str): returns length of C-string str char city[SIZE] = "Missoula"; cout << strlen(city); // prints 8</pre>
- strcat(str1, str2): appends str2 to the
 end of str1

```
char location[SIZE] = "Missoula, ";
char state[3] = "MT";
strcat(location, state);
// location now has "Missoula, MT"
```

Addison-Wesler is an imprint of

PEARSON

Copyright © 2015, 2012, 2009 Pearson Education, Inc., Publishing as Addison-Wesley All rights reserved

Library Functions for Working with C-Strings

Functions:

ostrcpy(str1, str2): copies str2 to str1
const int SIZE = 20;
char fname[SIZE] = "Maureen", name[SIZE];
strcpy(name, fname);

Note: strcat and strcpy perform no bounds checking to determine if there is enough space in receiving character array to hold the string it is being assigned.

Addison-Wesley is an imprint of

PEARSON

Copyright © 2015, 2012, 2009 Pearson Education, Inc., Publishing as Addison-Wesley All rights reserve

C-string Inside a C-string

Function:

ostrstr(str1, str2): finds the first occurrence of str2 in str1. Returns a pointer to match, or NULL if no match.

```
char river[] = "Wabash";
char word[] = "aba";
cout << strstr(state, word);
// displays "abash"</pre>
```

Addison-Wesle is an imprint o

PEARSON

Convirient © 2015-2012-2009 Pearson Education. Inc. Publishing as Addison-Wesley All rights reserve



10.5

C-String/Numeric Conversion Functions

ddison-Wesler is an imprint of

PEARSON

Copyright © 2015, 2012, 2009 Pearson Education, Inc., Publishing as Addison-Wesley All rights reserved

String/Numeric Conversion Functions

• Requires 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 2 nd parameter. 3 rd parameter is base of converted value

Addison-Wesla is an imprint of

PEARSON

Copyright © 2015, 2012, 2009 Pearson Education, Inc., Publishing as Addison-Wesley All rights reserved.

String/Numeric Conversion Functions

```
int iNum;
long lNum;
double dNum;
char intChar[10];
iNum = atoi("1234"); // puts 1234 in iNum
lNum = atol("5678"); // puts 5678 in lNum
dNum = atof("35.7"); // puts 35.7 in dNum
itoa(iNum, intChar, 8); // puts the string
// "2322" (base 8 for 123410) in intChar
```

String/Numeric Conversion Functions - Notes

- if C-string contains non-digits, results are undefined
 - ofunction may return result up to non-digit
 - ofunction may return 0
- itoa does no bounds checking make sure there is enough space to store the result

Addison-Wesley is an imprint of

PEARSON

Copyright © 2015, 2012, 2009 Pearson Education, Inc., Publishing as Addison-Wesley All rights reserved.



10.6

Writing Your Own C-String Handling Functions

ddison-Wesler s an imprint of

DEAD CON

Copyright © 2015, 2012, 2009 Pearson Education, Inc., Publishing as Addison-Wesley All rights reserved.

Writing Your Own C-String Handling Functions

- Designing C-String Handling Functions
 - ocan pass arrays or pointers to char arrays
 - Can perform bounds checking to ensure enough space for results
 - Can anticipate unexpected user input

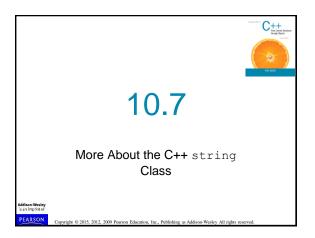
Addison-Wesler is an imprint of

PEARSON

Copyright © 2015, 2012, 2009 Pearson Education, Inc., Publishing as Addison-Wesley All rights reserved

From Program 10-9

From Program 10-10



The C++ string Class • Special data type supports working with strings • #include <string> • Can define string variables in programs: string firstName, lastName; • Can receive values with assignment operator: firstName = "George"; lastName = "Washington"; • Can be displayed via cout cout << firstName << " " << lastName;</pre>

Copyright © 2015, 2012, 2009 Pearson Education, Inc., Publishing as Addison-Wesley All rights

```
Input into a string Object

Use cin >> to read an item into a string:
    string firstName;
    cout << "Enter your first name: ";
    cin >> firstName;

Cin >> Copyright © 2015, 2012, 2009 Pearson Education, Inc., Publishing as Addison-Wesley All rights reserved.
```

• Use getline function to put a line of input, possibly including spaces, into a string: string address; cout << "Enter your address: "; getline(cin,address);

dison Wesley
at implies of

Input into a string Object

string Comparison

 Can use relational operators directly to compare string objects:

Comparison is performed similar to strcmp function.
 Result is true or false

Addison-Wesley is an imprint of

PEARSON Copyright © 2015, 2012, 2009 Pearson Education, Inc., Publishing as Addison-Wesley All rights

Other Definitions of C++ strings

Definition	Meaning
string name;	defines an empty string object
string myname("Chris");	defines a string and initializes it
string yourname(myname);	defines a string and initializes it
string aname(myname, 3);	defines a string and initializes it with first 3 characters of myname
string verb(myname, 3, 2);	defines a string and initializes it with 2 characters from myname starting at position 3
string noname('A', 5);	defines string and initializes it to 5 'A's

Addison-Wesley is an imprint of

PEARSO

Copyright © 2015, 2012, 2009 Pearson Education, Inc., Publishing as Addison-Wesley All rights reserved

string Operators

OPERATOR	MEANING	
>>	extracts characters from stream up to whitespace, insert into string	
<<	inserts string into stream	
=	assigns string on right to string object on left	
+=	appends string on right to end of contents on left	
+	concatenates two strings	
[]	references character in string using array notation	
>, >=, <, <=, ==, !=	relational operators for string comparison. Return true or false	

Addison-Wesh is an imprint of

PEARSON Copyright © 2015, 2012, 2009 Pearson Education, Inc., Publishing as Addison-Wesley All rights reserved

string Operators

string Member Functions

- Are behind many overloaded operators
- Categories:
 - oassignment: assign, copy, data
 - o modification: append, clear, erase, insert, replace, swap
 - space management: capacity, empty, length, resize, size
 - osubstrings: find, front, back, at, substr
 - ocomparison: compare
- See Table 10-8 for a list of functions.

Addison-Wesley is an imprint of

PEARSON

Copyright © 2015, 2012, 2009 Pearson Education, Inc., Publishing as Addison-Wesley All rights reserved

string Member Functions

Addison-Wesley is an imprint of

PEARSO

Copyright © 2015, 2012, 2009 Pearson Education, Inc., Publishing as Addison-Wesley All rights reserved.

```
string Member Functions in Program 10-21
```

Addison-Wesl is an imprint

PEARSO

Copyright © 2015, 2012, 2009 Pearson Education, Inc., Publishing as Addison-Wesley All rights reserve