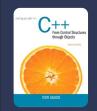
8 studiers in 3 days * 5.0 (1 review)

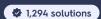
Textbook solutions for this set





Starting Out with C++ from Control Structures to Objects

8th Edition • ISBN: 9780133769395 (12 more) Godfrey Muganda, Judy Walters, Tony Gaddis





Sears and Zemansky's University Physics with Modern **Physics**

13th Edition • ISBN: 9780321696861 (1 more) Hugh D. Young, Lewis Ford, Roger A. Freedman

7,837 solutions

Search for a textbook or question >

- [1] What must I change in the test to go to the next iteration?
- [2] What information is produced?
- [3] What must I do to enter the loop?
- [4] Can my loop reach its bounds?
- [5] Has my loop reached its goal?
- [6] How is the data processed?
- [7] Can my loop be entered at all?
- [8] What makes this loop quit?

- [1] advance the loop
- [2] goal precondition
- [3] bounds precondition
- [4] necessary bounds
- [5] loop postcondition
- [6] loop operations and actions
- [7] loop guards

- [1] May not repeat its actions at all
- [2] Keeps processing input until a particular value is found in input.
- [3] Repeats its actions at least once
- [4] Keeps processing until the output gets no closer to the answer.
- [5] Test for the occurrence of a particular event
- [6] Repeats its actions a fixed number of times
- [7] Conditions under which a loop will repeat its actions
- [8] Keeps processing until the input device signals that it is finished.

- [8] loop bounds
- [1] guarded loop [2] sentinel loop
- [3] unguarded loop
- [4] limit loop
- [5] indefinite loop
- [6] definite loop
- [7] loop bounds
- [8] data loop

[1] Actions that occur after the loop is complete	[1] postcondition
[2] Actions occuring inside the loop's body	[2] operation
[3] Actions that occur before the loop is encountered	[3] precondition
[4] A test that determines if the loop should be entered	[4] bounds
	for (auto e : s)
Which of these is a flow-of-control statement?	if (x < 3) else
	while (x < 3)
Milkigh of these are grounded lease?	for
Which of these are guarded loops?	while
Which of these are unguarded loops?	do-while
	definite
Which are the two major categories of loops?	indefinite
	indefinite

Which of these are indefinite loops?	sentinel bounds limit bounds data bounds
Using the loop-building strategy from Chapter 5, which of these are part of the loop mechanics?	loop bounds bounds precondition advancing the loop
Look at the problem statement below. The of the loop is to count the number of characters in a sentence.	goal
[How many characters are in a sentence? Count the characters in a string until a period is encountered. If the string contains any characters, then it will contain a period. Count the period as well.]	
Look at the problem statement below. The of the loop is that a period was encountered.	bounds
[How many characters are in a sentence? Count the characters in a string until a period is encountered. If the string contains any characters, then it will contain a period. Count the period as well.]	
Look at the problem statement below. The of the loop is read a character and increment a counter.	plan
[How many characters are in a sentence? Count the characters in a string until a period is encountered. If the string contains any characters, then it will contain a period. Count the period as well.]	
Loop bounds used when searching through input.	sentinel bounds
	j sentinei sounus
Loop bounds often used in scientific and mathematical applications.	limit bounds
applications. In the classic for loop, loop control variables going from 0 to	limit bounds
applications. In the classic for loop, loop control variables going from 0 to less-than n are said to employ: Loop bounds used when reading files or processing network	limit bounds asymmetic bounds
In the classic for loop, loop control variables going from 0 to less-than n are said to employ: Loop bounds used when reading files or processing network data. How many times is this loop entered? (That is, how many	limit bounds asymmetic bounds data bounds
In the classic for loop, loop control variables going from 0 to less-than n are said to employ: Loop bounds used when reading files or processing network data. How many times is this loop entered? (That is, how many times is i printed?) for (int i = 1; i < 10; i++) cout << i;	limit bounds asymmetic bounds data bounds
In the classic for loop, loop control variables going from 0 to less-than n are said to employ: Loop bounds used when reading files or processing network data. How many times is this loop entered? (That is, how many times is i printed?) for (int i = 1; i < 10; i++) cout << i; cout << endl; How many times is this loop entered? (That is, how many	limit bounds asymmetic bounds data bounds 9

exp (part - In the dataset for book which portion of code is executed after the facts sharement in the book book? In the dataset for book which portion of code is not followed by semination of the facts of the fa	How many times is this loop entered? (That is, how many times is i printed?)	10
tine the classic for toda, within partition of code is not rottowed by a service of the today within partition of code is not rottowed by a service of the classic for today, which partition of code is executed after the last distance in the long body? In the classic for today, which partition of code is executed after the last distance in the long body? In the classic for today, which partition is used to create the last distance in the long body? In the classic for today, which partition is used to create the last distance in the long body? In the classic for today, which partition is used to create the last statement? In the classic for today, which partition is used to create the last statement? In the classic for today, which partition is used to create the last statement? In the classic for today, which partition is used to create the last statement? In the classic for today, which partition is used to create the last statement? In the classic for today, which partition is used to create the last statement? In the classic for today, which partition is used to create the last statement? In the classic for today, which partition is used to create the last statement? In the classic for today, which partition is used to create the last statement from the today building state used to create the counter variable, instaired to 1 Set counter to 00 While impression and correst-character and a period the partition is a period the counter variable. Since the rest character from or in commencement of the counter in section for the period. Fine Set counter to section for the period. Fine Set counter to 2 1 Ecourate is a fine earning was empty.	cout << i;	
courses can't sendil; In the classor for loop, which portion of code is analogous to attention of the code is consoled after the last statement in the cop body? In the classor for loop, which portion of code is analogous to an it statement in the cop body? In the classor for loop, which portion of code is analogous to an it statement? In the classor for loop, which portion is used to create the loop body analogous to an it statement? In the classor for loop, which portion is used to create the loop body which portion is used to create the loop corror variables of loop which portion is used to create the loop corror variables and it is a string (may be empty). Create the variable stril is a string (may be empty). Create the variable correct-character as a character Place the first character is string to current-character. While now characters and current-character. While now characters and current character not a period. Add one to the counter to a control to the period. The counter to 2 counter to 3 counter to 3 counter to 3 counter to 4 counter to 2 counter to 3 counter to 3 counter to 3 counter to 3 counter to 4 counter to 2 counter to 5 counter variable. If counter is 5 the string was empty.		11
In the classic for loop, which portion of code is executed after the last statement in the loop body? In the classic for loop, which portion of code is analogous to an if statement? In the classic for loop, which portion of code is analogous to an if statement? In the classic for loop, which portion of code is analogous to an if statement? In the classic for loop, which portion is used to create the copy control warrable? Because the allustration from the loop building strategy in Chapter 5. The highlighted lines represents Given the variable at it as a string (may be empty) Create the counter variable, initiatized to 1 Set counter to 0 Create the variable current character as a character. Place the first character in str indo current character. While more character in str in the counter variable. Store the rest character from str in current-character. If current character is a period then Add one to log increment) the counter to account for the period. Bits Store the rest character is a counter to account for the period. Bits Store the counter to account for the period. Bits Store the rest character is a period then Add one to the counter to account for the period. Bits Store the rest character is a period then Add one to the counter to account for the period. Bits Store the rest character is a period then Add one to the counter to account for the period. Bits Store the rest character is a period then Add one to the counter to account for the period. Bits Store the rest character is a period then Add one to the counter to account for the period. Bits Store the string was empty.	cout << i;	
In the classic for loop, which portion of code is analogous to an if statement? In the classic for loop, which portion is used to create the loop control variable? Below is the illustration from the loop building strategy in Chapter 5. The highlighted lines represents Given the variable str is a string frang be empty) Create the counter variable, initiatized to -1 If the variable str has any characters then the loop form of the variable str is a string frang be empty) Create the variable str is a string frang be empty) Create the variable str has any characters then the variable current-character as a character flace the first character in str into current character as a character flace the first character in str into current character and current character and current character and current character in str into current characte		update expression
In the classic for toop, which portion is used to create the toop control variable? Below is the illustration from the loop building strategy in Chapter 5. The highlighted lines represents Given the variable str is a string (may be empty) Create the counter variable, initialized to -1 ** If the variable str has any characters then ** Set counter to 0 Create the variable current-character as a character Place the first character in third current-character While more-characters and current-character not a period { Add one to (or increment) the counter variable Store the next character from str in current-character } If current-character is a period then Add one to the counter to account for the period. Else Set counter to -2 } If counter is -1 the string was empty		update expression
an it statement? In the classic for toop, which portion is used to create the toop control variable? Below is the illustration from the loop building strategy in Chapter 5. The highlighted lines represents Given the variable str is a string (may be empty) Create the counter variable, initialized to -1 ** If the variable str has any characters then ** (Set counter to 0 Create the variable current-character as a character Place the first character in all rinto current-character While more-characters and current-character not a period { Add one to (or increment) the counter variable Store the next character from str in current-character } If current-character is a period then Add one to the counter to account for the period. Else Set counter to -2 } If counter is -1 the string was empty		
Below is the illustration from the loop building strategy in Chapter 5. The highlighted lines represents: Given the variable str is a string (may be empty) Create the counter variable, initialized to -1 ## If the variable str has any characters then ## [Set counter to 0 Create the variable current-character as a character Place the first character in str into current-character While more-characters and current-character not a period { Add one to (or increment) the counter variable Store the next character from str in current-character } If current-character is a period then Add one to the counter to account for the period Esse Set counter to -2 } If counter is -1 the string was empty		condition expression
Chapter 5. The highlighted lines represents: Given: the variable str is a string (may be empty) Create the counter variable, initialized to -1 ## If the variable str has any characters then ## { Set counter to 0 Create the variable current-character as a character Place the first character in str into current-character While more-characters and current-character not a period { Add one to (or increment) the counter variable Store the next character from str in current-character } If current-character is a period then Add one to the counter to account for the period. Else Set counter to -2 } If counter is -1 the string was empty		initialization statement
Create the counter variable, initialized to -I # If the variable str has any characters then # { Set counter to 0 Create the variable current-character as a character Place the first character in str into current-character While more-characters and current-character not a period { Add one to (or increment) the counter variable Store the next character from str in current-character } If current-character is a period then Add one to the counter to account for the period. Else Set counter to -2 } If counter is -1 the string was empty		a loop guard
If the variable str has any characters then Set counter to 0 Create the variable current-character as a character Place the first character in str into current-character While more-characters and current-character not a period { Add one to (or increment) the counter variable Store the next character from str in current-character } If current-character is a period then Add one to the counter to account for the period. Else Set counter to -2 } If counter is -1 the string was empty		
Create the variable current-character as a character Place the first character in str into current-character While more-characters and current-character not a period { Add one to (or increment) the counter variable Store the next character from str in current-character } If current-character is a period then Add one to the counter to account for the period. Else Set counter to -2 } If counter is -1 the string was empty	If the variable str has any characters then	
Create the variable current-character as a character Place the first character in str into current-character While more-characters and current-character not a period { Add one to (or increment) the counter variable Store the next character from str in current-character } If current-character is a period then Add one to the counter to account for the period. Else Set counter to -2 } If counter is -1 the string was empty	(
period { Add one to (or increment) the counter variable Store the next character from str in current-character } If current-character is a period then Add one to the counter to account for the period. Else Set counter to -2 } If counter is -1 the string was empty	Create the variable current-character as a character	
Store the next character from str in current-character } If current-character is a period then Add one to the counter to account for the period. Else Set counter to -2 } If counter is -1 the string was empty		
Add one to the counter to account for the period. Else Set counter to -2 } If counter is -1 the string was empty		
	Add one to the counter to account for the period. Else	
	}	

Below is the illustration from the loop building strategy in goal precondition Chapter 5. The highlighted lines represents: Given: the variable str is a string (may be empty) Create the counter variable, initialized to -1 If the variable str has any characters then Set counter to 0 Create the variable current-character as a character Place the first character in str into current-character While more-characters and current-character not a period Add one to (or increment) the counter variable Store the next character from str in current-character If current-character is a period then Add one to the counter to account for the period. Else Set counter to -2 If counter is -1 the string was empty Else if counter is -2 there was no period Below is the illustration from the loop building strategy in bounds precondition Chapter 5. The highlighted lines represents:

Given: the variable str is a string (may be empty)

Create the counter variable, initialized to -1

If the variable str has any characters then
{
Set counter to 0

Create the variable current-character as a character
Place the first character in str into current-character

While more-characters and current-character not a period
{
Add one to (or increment) the counter variable
Store the next character from str in current-character
}

If current-character is a period then
Add one to the counter to account for the period.

Else
Set counter to -2

If counter is -1 the string was empty Else if counter is -2 there was no period ooungs precondition

a necessary condition The highlighted selection below illustrates: Given: the variable str is a string (may be empty) Create the counter variable, initialized to -1 If the variable str has any characters then Set counter to 0 Create the variable current-character as a character Place the first character in str into current-character While more-characters and current-character not a period Add one to (or increment) the counter variable Store the next character from str in current-character If current-character is a period then Add one to the counter to account for the period. Else Set counter to -2 If counter is -1 the string was empty Else if counter is -2 there was no period Below is the illustration from the loop building strategy in loop bounds

Chapter 5. The highlighted lines represents:

Given: the variable str is a string (may be empty)
Create the counter variable, initialized to -1

If the variable str has any characters then
{
Set counter to 0

Create the variable current-character as a character
Place the first character in str into current-character

While more-characters and current-character not a period

Create the variable current in str into current-character

If the variable current in str into current-character

If current-characters and current-character not a period

If current-character from str in current-character

If current-character is a period then

Add one to the counter to account for the period.

Else

Set counter to -2

If counter is -1 the string was empty Else if counter is -2 there was no period

The highlighted selection below illustrates: an intentional condition Given: the variable str is a string (may be empty) Create the counter variable, initialized to -1 If the variable str has any characters then Set counter to 0 Create the variable current-character as a character Place the first character in str into current-character While more-characters and current-character not a period Add one to (or increment) the counter variable Store the next character from str in current-character If current-character is a period then Add one to the counter to account for the period. Else Set counter to -2 If counter is -1 the string was empty Else if counter is -2 there was no period Below is the illustration from the loop building strategy in goal operation

Chapter 5. The highlighted lines represents:

Given: the variable str is a string (may be empty)

Create the counter variable, initialized to -1

If the variable str has any characters then
{
Set counter to 0

Create the variable current-character as a character
Place the first character in str into current-character

While more-characters and current-character not a period
{

*
Add one to (or increment) the counter variable

*
Store the next character from str in current-character
}

If current-character is a period then
Add one to the counter to account for the period.

Else
Set counter to -2

If counter is -1 the string was empty Else if counter is -2 there was no period oal operation

Below is the illustration from the loop building strategy in Chapter 5. The highlighted lines represents: Given: the variable str is a string (may be empty) Create the counter variable, initialized to -1 If the variable str has any characters then Set counter to 0 Create the variable current-character as a character Place the first character in str into current-character While more-characters and current-character not a period Add one to (or increment) the counter variable Store the next character from str in current-character If current-character is a period then Add one to the counter to account for the period. Else Set counter to -2 If counter is -1 the string was empty Else if counter is -2 there was no period

advancing the loop

Chapter 5. The highlighted lines represents: Given: the variable str is a string (may be empty) Create the counter variable, initialized to -1 If the variable str has any characters then { Set counter to 0 Create the variable current-character as a character Place the first character in str into current-character While more-characters and current-character not a period { Add one to (or increment) the counter variable Store the next character from str in current-character } ## If current-character is a period then ## Add one to the counter to account for the period. Else Set counter to -2]

If counter is -1 the string was empty Else if counter is -2 there was no period

Below is the illustration from the loop building strategy in

loop postcondition

In a guarded loop, the loop actions may never be executed	True
In a guarded loop, the loop actions are always executed at	
least once.	False
In an unguarded loop, the loop actions are always executed at least once.	True
In an unguarded loop, the loop actions may never be executed.	False
A guarded loop is also known as a test-at-the-top loop	True
A guarded loop is also known as a test-at-the-bottom loop.	False
An unguarded loop is also known as a test-at-the-bottom loop.	True
An unguarded loop is also known as a test-at-the-top loop.	False
Loops are used to implement iteration in C++.	True
Loops are used to implement selection in C++.	False
This idiomatic pattern is used to count from one value to another.	True
for (int i = 1; i <= 10; i++) cout << i; cout << endl;	
This idiomatic pattern is used to count from one value to another.	
for (int i = 1; i < 10; i++)	
cout << i; cout << endl;	False
This loop uses asymmetric bounds.	True
for (int i = 0; i < 10; i++)	
cout << i; cout << endl;	
This loop uses asymmetric bounds.	True
for (int i = 1; i < 10; i++)	
cout << i; cout << endl;	
This loop uses asymmetric bounds.	
for (int i = 1; i <= 10; i++)	
cout << i; cout << endl;	
	False