CH 05 QUIZ

Share

8 studiers in 3 days ★ 5.0 (1 review)

Terms in this set (40)

[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 [8] loop bounds
[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.	[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 [2] Actions occuring inside the loop's body [3] Actions that occur before the loop is encountered [4] A test that determines if the loop should be entered	[1] postcondition [2] operation [3] precondition [4] bounds

Which of these is a flow-of-control statement?	for (auto e : s) ...
	
	if (x < 3) ... else ...
	
	while (x < 3) ...

Which of these are guarded loops?	for
	while

Which of these are unguarded loops?	do-while
Which are the two major categories of loops?	definite 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. [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.]	goal
Look at the problem statement below. The ______ of the loop is that a period was encountered. [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.]	bounds
Look at the problem statement below. The ______ of the loop is read a character and increment a counter. [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.]	plan

Loop bounds used when searching through input.	sentinel bounds
Loop bounds often used in scientific and mathematical applications.	limit bounds
In the classic for loop, loop control variables going from 0 to less-than n are said to employ:	asymmetic bounds
Loop bounds used when reading files or processing network data.	data bounds

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;
	9

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;
	10

How many times is this loop entered? (That is, how many times is i printed?)
for (int i = 0; i < 10; i++)
cout << i;
cout << endl;
	10

How many times is this loop entered? (That is, how many times is i printed?)
for (int i = 0; i <= 10; i++)
cout << i;
cout << endl;
	11

In the classic for loop, which portion of code is not followed by a semicolon?	update expression
In the classic for loop, which portion of code is executed after the last statement in the loop body?	update expression
In the classic for loop, which portion of code is analogous to an if statement?	condition expression
In the classic for loop, which portion is used to create the loop control variable?	initialization statement

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
	a loop guard

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
```

goal precondition

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
```

bounds precondition

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
```

a necessary condition

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
```

loop bounds

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
```

an intentional condition

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
```

goal 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

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
```

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.
for (int i = 1; i <= 10; i++)
cout << i;
cout << endl;
	True

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.
for (int i = 0; i < 10; i++)
cout << i;
cout << endl;
	True

This loop uses asymmetric bounds.
for (int i = 1; i < 10; i++)
cout << i;
cout << endl;
	True

This loop uses asymmetric bounds.
for (int i = 1; i <= 10; i++)
cout << i;
cout << endl;
	False