Due no later than 9:00pm on Sunday 10/10 – Submit via Cougar Courses

Note: Material from these OLI pages is NOT included in Test 1!

As you read the following OLI pages and complete the interactive activities, capture the screenshots of the completed activities and replace the respective screenshots in the document.

- Page 25 Functions to display
- Page 26 Pass-by-value parameters
- Page 27 Pass-by-reference

When you are ready to submit the assignment, download the document in PDF and submit the PDF file on Cougar Course as the proof for your work.

Page 25 functions to display

LBD who says what

Make a function call in the program so that it will display:

This is a dog.

It says woof.

When you are successful, choose the appropriate arguments in the following function call to reflect what you did.

print_who_says_what("dog" , "woof");

✓ Correct. The first argument is for the animal parameter.

✓ Correct. The second argument is for the sound parameter.

Modify the output statements in the ***function body*** of `print_who_says_what` to simplify the message.

For example, when we call `print_who_says_what("Cow", "moo")` the message will say

Cow goes moo!

When we call `print_who_says_what("Cat", "meow")`, the message will say

Cat goes meow!

Run the program to make sure it's working properly. Enter the new output statement below.

```
cout << animal << " goes " << sound << ".\n";
```

Resubmit

✓ cout << animal << " goes " << sound << "!\n";

Add the following statement inside the main function.

```
cout << animal;
```

Run the program to see the error message generated by the compiler. Why do you think the compiler is not happy about this statement?

Because that variable isn't in the main , but it's specific function.

Resubmit

✓ The parameter animal is declared inside the function header of print_who_says_what. It is only accessible inside the print_who_says_what function and not accessible inside the main function.

```
string pet, pet_sound;
cout << "What is your pet?\t";
getline(cin, pet);
cout << "What sound does it make?\t";
getline(cin, pet_sound);
```

Given the above code segment. Which of the following function call could be added after it to display what the pet says?

○ print_who_says_what("pet", "pet_sound");

[✓ print_who_says_what(pet, pet_sound);]

○ print_who_says_what(pet_sound, pet);

○ print_who_says_what(animal, sound);

✓ Correct, this will use the value stored in pet as the argument for the animal parameter and the value stored in pet_sound as the argument for the sound parameter.

LBD expected behavior of print_ft_in

What is the expected outcome for `print_ft_in(80)`? 6 ft 8 in

✓ Correct! 80 / 12 is 6 and 80 % 12 is 8.

What argument should be used to get **`4 ft 10 in`** as the expected outcome?

`print_ft_in(` 58 **`)`**

✓ Correct!

LBD calling print_ft_in

However, the main function is missing a call to the print_ft_in function to display the user's height in feet and inches.

Which of the following should be added to the main function in order to display the user's height in feet and inches?

- ○ cout << print_ft_in(my_height);
- [✓ print_ft_in(my_height);]
- ○ print_ft_in(total_inches);

✓ Correct; this print_ft_in function does not have a return value and must be called as a stand-alone statement.

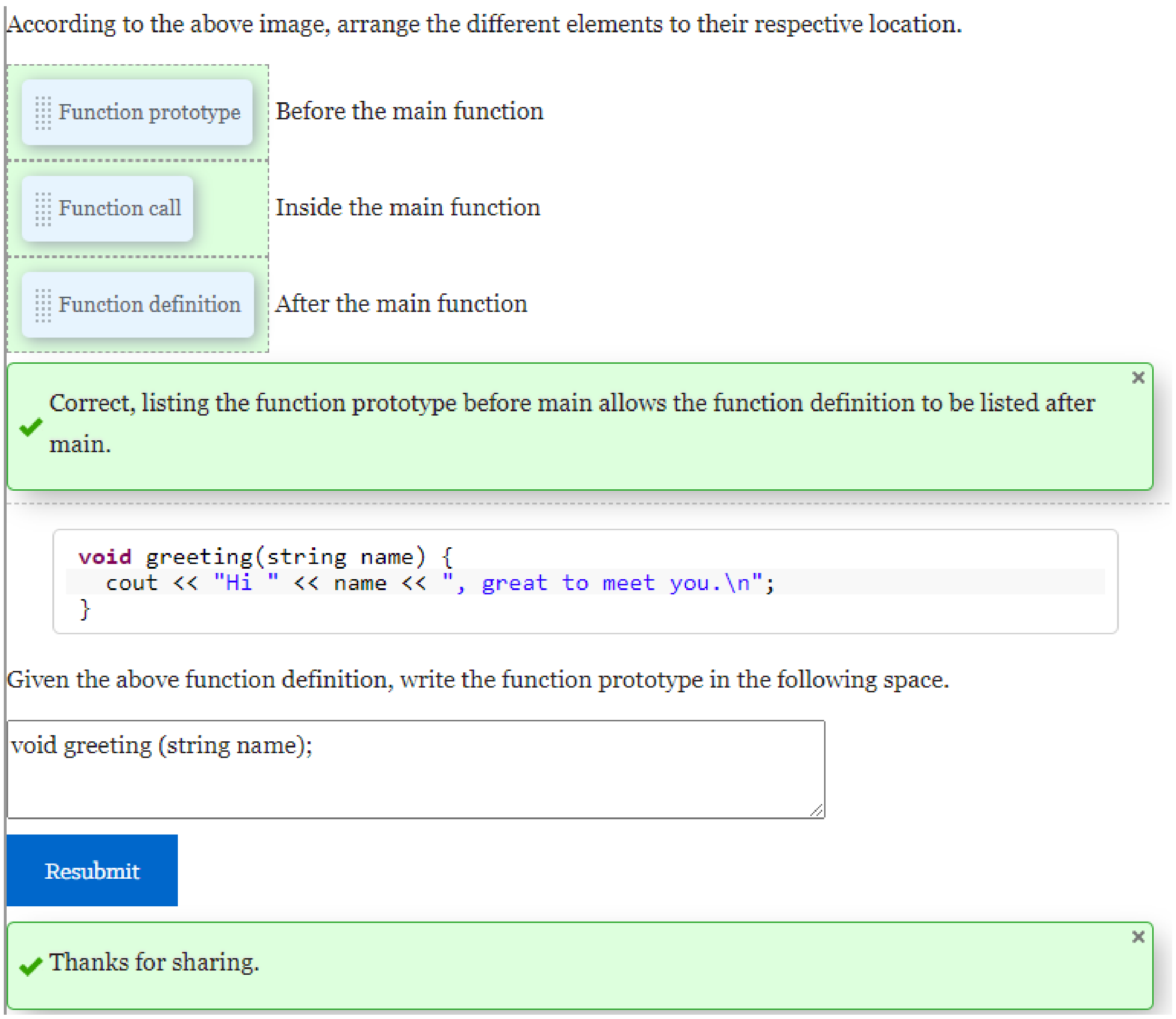
What questions or tips for others do you have regarding functions with void at the beginning of its function header?

None

Resubmit

✓ Thanks for sharing.

Hotspot function prototype, definition, call



According to the above image, arrange the different elements to their respective location.

Function prototype	Before the main function
Function call	Inside the main function
Function definition	After the main function

✓ Correct, listing the function prototype before main allows the function definition to be listed after main.

```
void greeting(string name) {
   cout << "Hi " << name << ", great to meet you.\n";
}
```

Given the above function definition, write the function prototype in the following space.

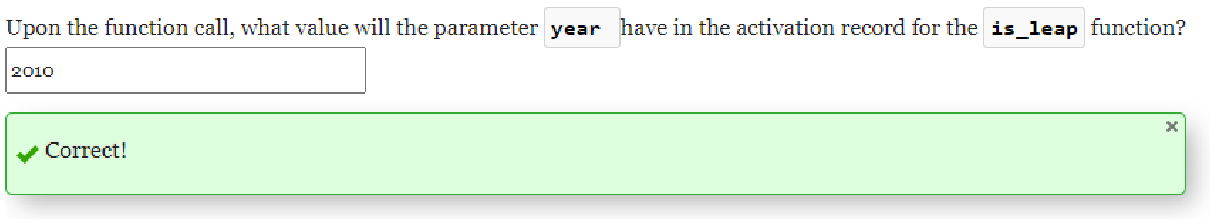
void greeting (string name);

Resubmit

✓ Thanks for sharing.

Page 26 pass-by-value parameters

LBD leap year

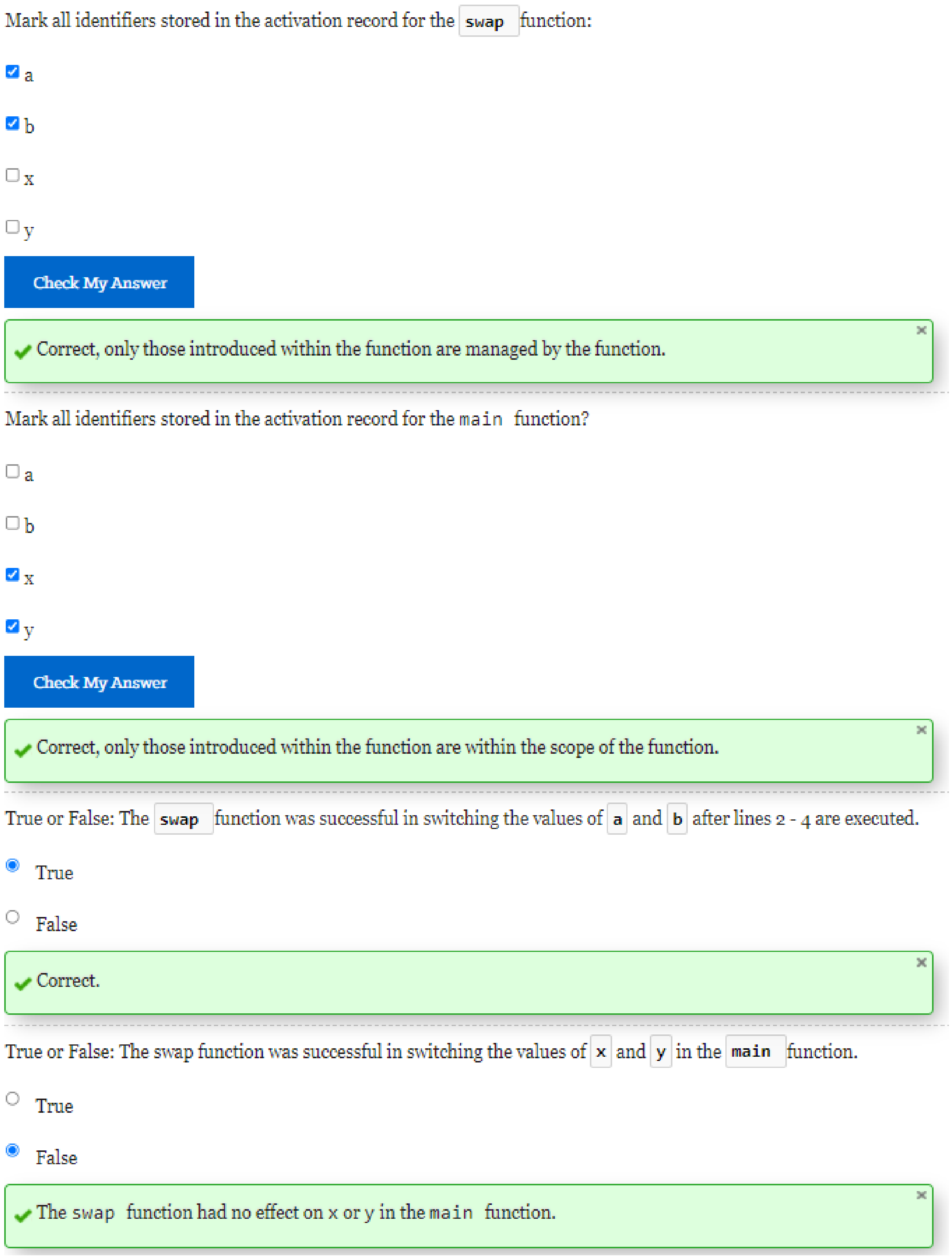


Upon the function call, what value will the parameter `year` have in the activation record for the `is_leap` function?

2010

✓ Correct!

LBD limitation with pass-by-value



Mark all identifiers stored in the activation record for the `swap` function:

[✓ a]

[✓ b]

x

y

Check My Answer

✓ Correct, only those introduced within the function are managed by the function.

Mark all identifiers stored in the activation record for the `main` function?

a

b

[✓ x]

[✓ y]

Check My Answer

✓ Correct, only those introduced within the function are within the scope of the function.

True or False: The `swap` function was successful in switching the values of `a` and `b` after lines 2 - 4 are executed.

[✓ True]

False

✓ Correct.

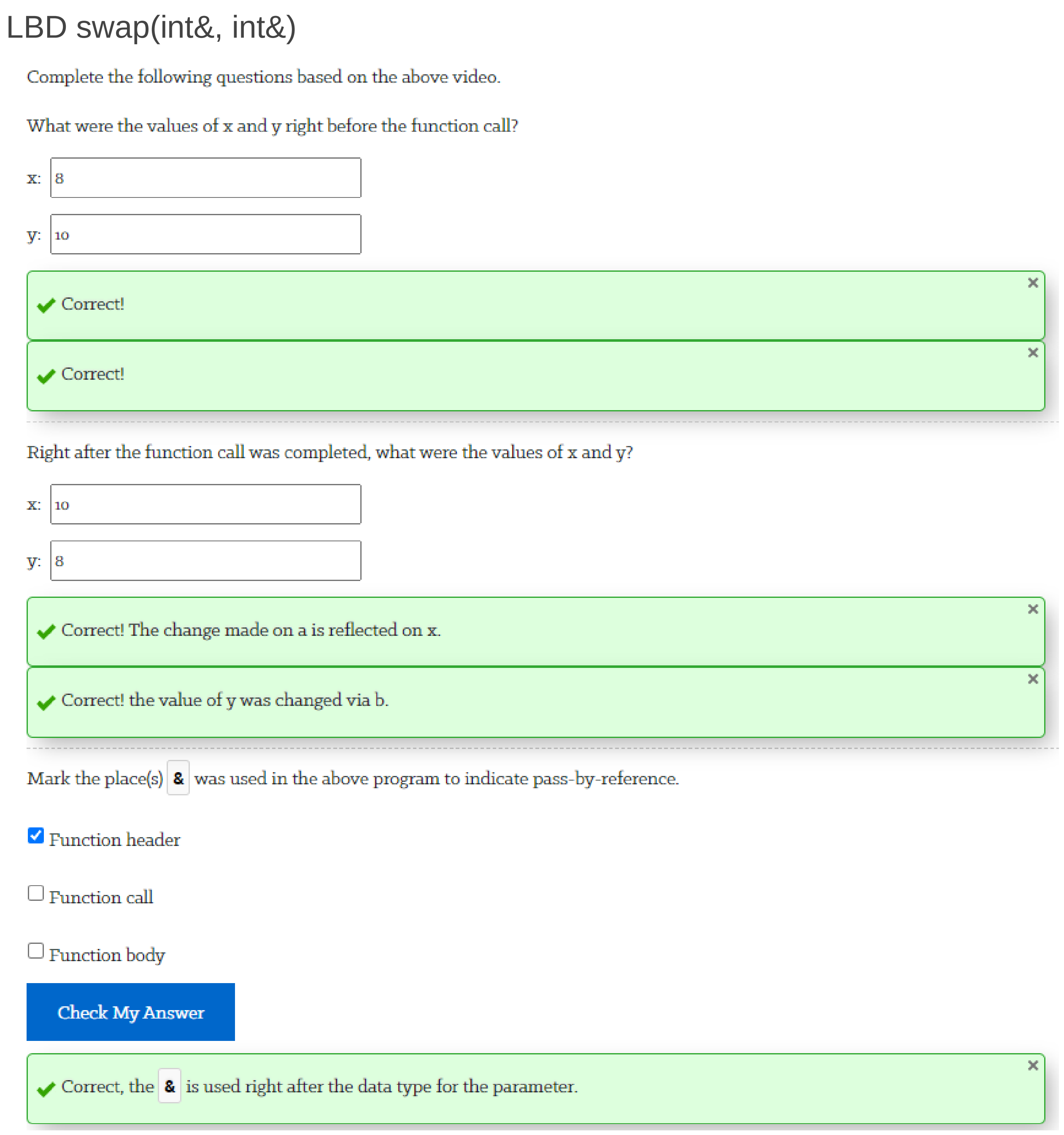
True or False: The swap function was successful in switching the values of `x` and `y` in the `main` function.

True

[✓ False]

✓ The `swap` function had no effect on x or y in the `main` function.

Page 27 Pass-by-reference parameters



LBD swap(int&, int&)

Complete the following questions based on the above video.

What were the values of x and y right before the function call?

x: 8

y: 10

✓ Correct!

✓ Correct!

Right after the function call was completed, what were the values of x and y?

x: 10

y: 8

✓ Correct! The change made on a is reflected on x.

✓ Correct! the value of y was changed via b.

Mark the place(s) `&` was used in the above program to indicate pass-by-reference.

[✓ Function header]

Function call

Function body

✓ Correct, the `&` is used right after the data type for the parameter.

LBD collect_package_info

```
collect_package_info (first_package_cost, _______);
```

○ second_package_quantity

○ quantity

[✓ first_package_quantity]

✓ Correct, this would allow the function to store the value collected for quantity to be stored in first_package_quantity. ×

What do you think the above function call is doing?

I think it'll output a response accordingly to the first package cost and first package quantity

Resubmit

What should be used to replace the second argument in the function call

```
collect_package_info (_______, second_package_quantity);
```

○ cost

○ second_package_quantity

○

○ second_unit_cost

[✓ second_quantity_cost]

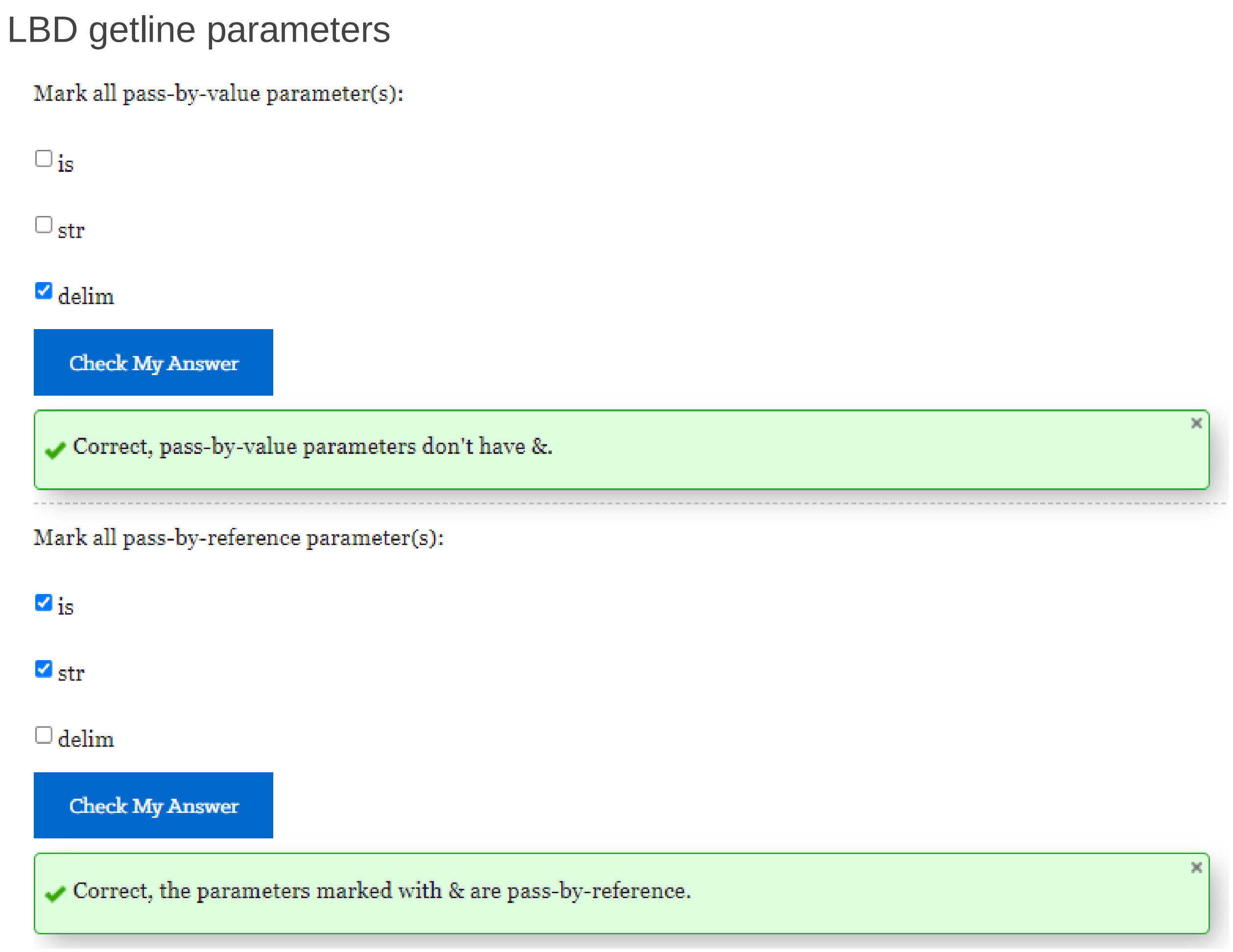
✓ Correct, this would allow the function to store the value collected for quantity to be stored in first_package_quantity. ×

What questions or tips do you have regarding the collect_package_info function.

None

Resubmit

✓ Thanks for sharing. ×



LBD getline parameters

Mark all pass-by-value parameter(s):

is

str

[✓ delim]

Check My Answer

✓ Correct, pass-by-value parameters don't have &.

Mark all pass-by-reference parameter(s):

[✓ is]

[✓ str]

delim

Check My Answer

✓ Correct, the parameters marked with & are pass-by-reference.

Which of the following function calls would be valid in the `main` function?

```
parse_time(str, hour, minute);
```

[✓ Invalid]

Valid

✓ Correct; the three variables have been defined in the `main` function and can serve as valid arguments for the function call.

```
parse_time(time_stamp, parsed_hour, parsed_minute);
```

[✓ Valid]

Invalid

✓ Correct; the three variables have been defined in the `main` function and can serve as valid arguments for the function call.

```
parse_time("11:25", parsed_hour, parsed_minute);
```

[✓ Valid]

Invalid

✓ Correct, it's okay to use a value string literal since str is pass-by-value.

```
parse_time(time_stamp, 10, parsed_minute);
```

Valid

[✓ Invalid]

✓ Correct, the second parameter is ***pass-by-reference*** and needs to have a variable of the same data type as its argument.