Make a copy of this Google Doc template and save the copy in your Google Drive.

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 21 Choose from two options
- Page 22 Choose from multiple options
- Page 23 Logic/Boolean operators

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 21 Choose from two options (Part II)

LBD calling absolute

Mark the function calls that would trigger the execution of the *if-block* in the `absolute` function.

- [ ] `absolute(-28)`
- [x] `absolute(0)`
- [ ] `absolute(-4.1)`
- [x] `absolute(2.5)`

Check My Answer

✔ Correct! The if-block is executed when the parameter receives a non-negative value as its argument.

Complete the following function call with a value that would trigger the execution of the else-block (line 5) in the `absolute` function.

```
absolute( -1 );
```

✔ Correct! The else-block is executed when the parameter receives a negative value as its argument.

LBD if without else

Mark all of the following function calls that would trigger the ***if-block*** of the function body of `final_cost` to be executed:

- [x] final_cost(55)
- [ ] final_cost(45)
- [x] final_cost(50)

Check My Answer

✔ Correct, the if-blocked is executed whenever the purchase parameter takes on an argument that is 50 or more.

Enter a number in the following function call that would trigger the if-block in the function body to be executed:

leftover > 0

Correct, we only need one more box when the number of cupcakes are not divisible by the box size.

Mark all of the following function calls that would trigger the if-block of the function body of `get_boxes` to be executed:

- [x] get_boxes(8, 5)
- [x] get_boxes(10, 4)
- [ ] get_boxes(10, 5)
- [ ] get_boxes(8, 4)

Check My Answer

Correct, the if-blocked is executed whenever the argument for the cupcakes parameter is not divisible by the argument for the box_size parameter.

Describe a scenario when extra action is needed only when a certain condition is met.

Buying groceries

Resubmit

Page 22 Choose from multiple options

LBD calling water_state

Which assignment statement will be executed with the following function call?

state = "Solid"; // line 4

```
water_state(-10)
```

✔ Correct. Line 4 is executed when temperature <= 0 is true.

Which of function call would trigger the execution of the assignment in line 6?

water_state(25)

✔ Correct. Line 6 is executed when temperature <= 0 is false while temperature < 100 is true.

Enter an argument for the following function call in order for the return value of the function to be "Gas":

water_state(101)

✔ Correct! Anything greater than 100 would trigger the execution of line 8.

Mark all comparisons the compiler will need to evaluate before returning 'C' for the function call letter_grade(75).

- [ ] grade >= 60
- [x] grade >= 90
- [x] grade >= 70
- [x] grade >= 80

Check My Answer

Correct, the compiler will first evaluate grade >= 90, then grade >= 80, and finally grade >= 70. grade >= 60 will not be evaluated since grade >= 70 is true.

In your own word, explain why the function call `letter_grade(75)` would not return 'D' even though grade >= 60 is true.

Because 75 is greater than or equal to 70 which outputs the letter C.

Resubmit

Thanks for sharing.

What would be the problem if we change the function, using if instead of else if for the comparisons after grade >= 90? Click on this link to test the function in Repl.it.

```
char letter_grade(int grade) {
  char letter;
  if (grade >= 90) {
    letter = 'A';
  }
  if (grade >= 80) {
    letter = 'B';
  }
  if (grade >= 70) {
    letter = 'C';
  }
  if (grade >= 60) {
    letter = 'D';
  }
  else {
    letter = 'F';
  }
  return letter;
}
```

It uses the last if statement as the main so any input greater than or equal to 60 will output D, even if its 70+

Resubmit

LBD temperature message

What would the above function return for the function call `temp_message(89)`?

"Hot"

✔ Correct. Since 89 > 85 is true, the function assigns "Hot" to message and skip the rest of the comparisons and assignments.

What would the above function return for the function call `temp_message(105)`?

"Hot"

✔ Correct. Since 105 > 85 is true, the function assigns "Hot" to message and skip the rest of the comparisons and assignments.

What is another number you would use in the following function call to show that the the function is not running properly?

temp_message(1000)

✔ Correct! Since `temperature > 85` evaluates to `true`, the program will skip `else if (temperature >100)` and never assign "Very Hot" as intended.

Which of the following functions calls would return "Comfortable"?

- (●) temp_message(65)
- ( ) temp_message(44)
- ( ) temp_message(93)

✔ Correct. Any temperature in the range of (60, 85] would return "Comfortable".

MR consecutive range examples

What other examples of consecutive ranges can you think of?

How far you swing a golf ball

Resubmit

✔ Thanks for sharing. ×

What questions do you have regarding if-else if-else in general or the above examples?

None

Resubmit

✔ Thanks for sharing. ×

Checkpoint switch statement:

Copy and Paste a screenshot of your expanded switch statement to replace the following.

1. Make sure you are logged in with your Repl.it account.
2. Click this link to access a program that contains the above ice cream flavor survey.
3. Add your name and section on the top of the program.
4. Expand the switch statement to include additional cases so that each ice cream flavor option will receive its own feedback.

When you are done, copy and paste the switch statement into the following box.

```
switch (response) {
case 'A':
case 'a':
 cout << "Great, that's my favorite too!" << endl;
 break;
case 'B':
case 'b':
 cout << "Good choice!" << endl;
 break;
case 'C':
case 'c':
 cout << "Yummy!" << endl;\
 break;
case 'D':
case 'd':
 cout << "Flavorful Fruit!" << endl;
default:
 cout << "Sorry, not a valid option." << endl;
```

Resubmit

✔ Thanks for sharing.

Describe a scenario where different actions would be performed based on **specific** characters or int values.

Making a choice.

Resubmit

✔ Thanks for sharing.

Page 23 Logic/Boolean operators

LBD logic operator &&

Choose appropriate values for the following conditions using data from the above table.

row #	`sales >= 21000`	`sales < 23000`	`sales >= 21000 && sales < 23000`
3	false	true	false
4	true	false	false
6	true	true	true

- ✔ Correct. `16806 >= 22000` is `false`
- ✔ Correct. Since the comparison on the left of the `&&` operator is `false`, the whole expression is `false`.
- ✔ Correct, `23138 <= 22999` is `false`.
- ✔ Correct. Since the comparison on the right of the `&&` operator is `false`, the whole expression is `false`.
- ✔ Correct, both operands of the and operator evaluate to `true`.

Enter numbers of your choice to the first column of the following table to generate the results in the second and third columns. Choose the right value for the last column.

sales	`sales >= 21000`	`sales < 23000`	`sales >= 21000 and sales < 23000`
20000	false	true	false
24000	true	false	false
22000	true	true	true

- ✔ Correct! This would make `sales >= 21000` evaluate to `false` and `sales < 23000` evaluate to `true`.
- ✔ Correct, because `sales >= 21000` is `false`.
- ✔ Correct, because `sales < 23000` evaluate to `false`.
- ✔ Correct, because both `sales >= 21000` and `sales < 23000` evaluate to `true`.
- ✔ Correct! This would make `sales >= 23000` evaluate to `true` but `sales < 21000` evaluate to `false`.
- ✔ Correct! This would make `sales >= 21000` as well as `sales < 23000` evaluate to `true`.

The above function is set up to return true if hour is within the range of 1 and 12. In the following space, write the conditional expression using &&.

```
if (valid >= 1 && valid <= 12)
```

Resubmit

LBD logic operator ||

Choose appropriate values for the following conditions based on the value of my_grade in the first column.

`my_grade`	`my_grade < 0`	`my_grade > 100`	`my_grade <0 || my_grade > 100`
150	false	true	true
-96	true	false	true
85	false	false	false

✔ Correct. `150 < 0` is `false`

✔ Correct. Since the comparison on the left of the `||` operator is `true`, the whole expression is `true`.

✔ Correct, `-96 > 100` is `false`.

✔ Correct. Since the comparison on the left of the `||` operator is `true`, the whole expression is `true`.

✔ Correct, both operands of the `||` operator evaluate to `false`.

In your own words, explain why it is not a good idea to use (my_grade < 0 && my_grade > 100) as a conditional expression.

Because it is impossible to have a number less than 0 and greater than 100

Resubmit

✔ This would require my_grade to be both negative and larger than 100, which is impossible.

In your own words, explain why it is not a good idea to use (my_grade >= 0 || my_grade <= 100) as a conditional expression.

Because the condition will always be true

Resubmit

✔ This condition will never become false.

The above function is set up to return `false` if `hour` is outside the range of 1 and 12. Note the if-block sets `valid` to `false` and the else-block sets `valid` to `true`.

In the following space, write the conditional expression using `||`.

```
if (valid < 1 || valid > 12)
```

Resubmit

Do you have a preference between using `&&` to check for within the range and using `||` to check for outside the range?

```
Yes, I prefer || to check
```

Resubmit

✔ Thanks for sharing.

LBD logic operator !

Complete the following table with appropriate `bool` values.

`valid_hour(12)`	`!valid_hour(12)`	`valid_hour(12) == false`
true	false	false

10

✓ Correct, `!valid_hour(12)` has the opposite value to `valid_hour(12).`

✓ Correct, `valid_hour(12) == false` is the same as `!valid_hour(12)`.

We know that `valid_hour(20)` would return `false` because `20` is not between `1` and `12`. Complete the following table with appropriate `bool` values.

`valid_hour(20)`	`!valid_hour(20)`	`valid_hour(20) == false`
false	true	false

✓ Correct, `!valid_hour(20)` has the opposite value to `valid_hour(20).`

✓ Incorrect, `valid_hour(20) == false` is the same as `!valid_hour(20)`.

```
int value;
cout << "Enter a number representing an hour value:";
cin >> value;
bool checked = valid_hour(value)
```

The above code segment prompts for a user input and assigns the result of `valid_hour(value)` to a `bool` variable `checked`. Complete the following table with appropriate `bool` values according to the value of `checked`.

checked	!checked	checked == false
false	true	true
true	false	false

✓ Correct, when `checked` holds a bool value, `!checked` is always the opposite of `checked.`

✓ Correct, when `checked` holds a bool value, `!checked` is always the opposite of `checked.`

✓ Correct, when `checked` holds a `bool` value, `checked == false` is always the same as `!checked.`

✓ Correct, when `checked` holds a `bool` value, `checked == false` is always the same as `!checked.`

LBD Magic 8 ball

The find method returns `npos` if the given substring is not found in the invoking object. That is, `question.find("ball")` will return `npos` if `"ball"` is not found in `question`.

```
question.find("ball") != npos && question.find("we") != npos
```

Suppose the `magic_8` function now uses the above for its condition expression. Mark all function calls that would return "Yes".

- [x] magic_8("Can we play a ball game?")
- [ ] magic_8("Can we play the Magic 8 game?")
- [x] magic_8("Are we buying a Magic 8 ball app?")
- [ ] magic_8("Do I need a Magic 8 ball app?")

Check My Answer

✔ Correct, both of these questions contain "we" and "ball".

```
question.find("ball") != npos || question.find("we") != npos
```

Suppose the `magic_8` function now uses the above for its condition expression. Mark all function calls that would return "Yes".

- [x] magic_8("Can we play the Magic 8 game?")
- [x] magic_8("Can we play a ball game?")
- [x] magic_8("Do I need a Magic 8 ball app?")
- [x] magic_8("Are we buying a Magic 8 ball app?")

Check My Answer

✔ Correct, all of these questions contain either "we" or "ball".

In the following space, create your own conditional expression to replace the one in magic_8. Be sure to include at least one logic operator.

```
answer.find("May" != npos || answer.find("please") != npos
```

Resubmit

✔ Thanks for sharing.