## Congratulations! You passed!

**Grade received** 83.33% **Latest Submission Grade** 83.33% **To pass** 80% or higher

Go to next item

1.	Which of the following statements is <i>false</i> about Graph approach?	1/1 point
	O Faster compilation	
	Easier debugging	
	OPortability	
	O Parallelism	
	<ul> <li>Correct         Correct! This statement is false. Since operations don't execute until the Graph is fully designed, it can be tricky to debug.     </li> </ul>	
2.	Which of the following statements is <i>true</i> for <i>tf.cond</i> ?	1/1 point
	Graph execution does not support if/else statements. To replicate that effect you use tf.cond	
	tf.cond is an alternative to using if/else statements in Graphs, as its execution is much faster than if/else statements.	

3. Consider the following code:

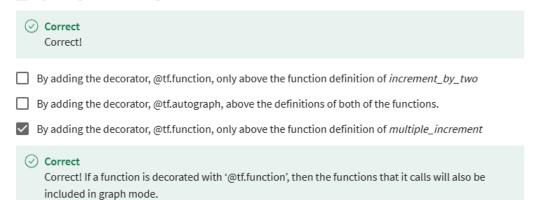
```
1/1 point
```

```
def increment_by_two(x):
    return x + 2

def multiple_increment(x, i):
    k = x
    for j in range(i):
        k = increment_by_two(k)
    return k
```

How do you convert both of these functions to execute in  ${\it Graph}$  mode? Check all that are true.

By adding the decorator, @tf.function, above the definitions of both of the functions.



4.	Function written in Eager mode when converted to Graph accommodates different data types all in one, so you don't have to define similar functions for different data types.	1/1 point
	True False	
	○ Correct     Correct!	
_	Which of the following is the country to the disclaration of the country of the c	
5.	Which of the following is the correct syntax to display the auto-generated AutoGraph code if your function name is <i>my_function</i> ?	0 / 1 point
	tf.autograph.code(my_function.python_function)  tf.autograph.to_code(my_function)	
	tf.autograph.to_code(my_function)  tf.autograph.to_code(my_function.python_function)	
	tf.autograph.code(my_function)	



O Hello World!

Hello World!

Hello World!

O Hello World!

## ✓ Correct

Correct! Even though tf.print is used, we still get 6 print statements because the function is not decorated to run as a Graph.