1/1 point

1/1 point

AutoGraph

100%	
Wh	ich of the following statements is false about Graph approach?
0	Faster compilation
0	Parallelism
	Wh





3. Consider the following code



w do you convert *both* of these functions to execute in *Graph* mode? Check all that are true.

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

By adding the decorator, @tf.function, only above the function definition of multiple_increment

```
Correct! If a function is decorated with '@tf.function', then the functions that it calls will also be included in graph mode.
By adding the decorator, @tf.function, only above the function definition of increment_by_two
```

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

```
✓ Correct!
```

 Function written in Eager mode when converted to to define similar functions for different data types. verted to Graph accommodates different data types all in one, so you don't have 1/1 point True

```
○ False
```

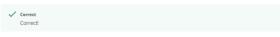
5. Which of the following is the correct syntax to display the auto-generated AutoGraph code if your function name is my_function?



 tf.autograph.to_code(my_function.python_function) (tf.autograph.code(my_function)

(tf.autograph.code(my_function.python_function)

_____ tf.autograph.to_code(my_function)



6. Consider the following code, what will be the output?



```
O Hello World!
    Hello World!
Hello World!
    Hello World!
    Hello World!
    Hello World!
    Hello World!
○ Hello World!
    Hello World!
    Hello World!
```

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