Learning Quiz 24: Modules and Classes as Namespaces

Due Nov 13 at 1pm **Points** 5 **Questions** 5 **Available** until Dec 4 at 11:59pm **Time Limit** None
Allowed Attempts Unlimited

Instructions

Prior to completing this quiz, be sure to read:

- Sections 7.4-7.5: Modules and Classes as Namespaces (p. 223-232)

Please also go over Practice Problems 7.5-7.6 in the textbook (solutions at the end of the chapter) before attempting this quiz.

This quiz was created for learning purposes. You may attempt this quiz as many times as you would like. The highest score *prior to the deadline*will count towards the final course grade. No late submissions will be accepted.

Take the Quiz Again

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	13 minutes	4 out of 5

Score for this attempt: **4** out of 5
Submitted Nov 10 at 7:25pm
This attempt took 13 minutes.

Question 1

1 / 1 pts

Early in our course, we learned to import the math and Fractions library. When we import these libraries, we are importing a package of functions and variables that belong to a different namespace. We, therefore, cannot use a variable pi directly (unless we created our own). However, we can use something like math.pi:

```
import math
print(math.pi)
print(math.sqrt(15))
print(pi)        ## This gives us an error
print(sqrt(15))  ## This also gives an error
```

Rather than referring to the libraries by their name, we can import with nicknames using the reserved word 'as'. Below, we import math and refer to math as m:

```
import math as m
print(m.pi)
```

Another special math constant is the number e, which equals 2.71828. If we started with the following code, how can we access the number e?

```
import math as mth
```

Correct!

[✓ mth.e]

e

math.e

m.e

Question 2

0.33 / 1 pts

We can create our own modules to import as a regular .py file. Download the following q23mod.py file and save it to your working directory in PyCharm (the same folder as your current .py file). You can also create an entirely separate .py file with the following lines of code:

```
def f():
    'prints we are running f()'
    print('We are running f()')

def g():
    'prints we are running g()'
    print('We are running g()')
```

We'll refer to this file as q23mod. To import it, type:

```
import q23mod
```

We can now use functions f() and g() in q23mod by typing:

```
q23mod.f()
q23mod.g()
```

Note, that unless you created your own f() and g() functions, f() and g() themselves won't work.

```
import q23mod

q23mod.f()
q23mod.g()
f()     # no such function f()
g()     # no such function g()
```

Using the q23mod.py file, consider the following lines of code:

```
import q23mod as q23

def h():
    'prints we are running h()'
    print('We are running h()')
```

Which of the following functions can we use?

Correct!

[✓ q23.g()]

[ ] f()

[ ] q23.h()

Correct! [✓ h()]

You Answered [✓ q23mod.f()]

Correct! [✓ q23.f()]

You Answered [✓ q23mod.g()]

[ ] q23mod.h()

[ ] g()

Question 3

1 / 1 pts

If our file is in a *different folder*, we will first need to tell Python where to look. We can do this by entering the following lines of code before importing the desired module:

```
import sys
sys.path.append('/Users/Name/Documents')     ## Change the path to match your computer

import q23mod
```

Note that assumes that my file q23mod.py is in my Documents folder. Your path WILL look different from this. Try moving the q23mod.py file to a different folder and check to make sure you are able to import the module. Confirm that you were able to import the module from a different folder.

Correct! [✓ I confirm that I successfully imported the module from a different folder (and/or I have made several attempts and I will message the instructor for assistance)]

( ) I did not attempt to complete this task

Question 4

0.67 / 1 pts

To import just a particular function from a module, you can use the **from** command. See, for example:

```
from q23mod import f
```

By using the from-import sequence, we do not need to prefix the function with the module name. That is, we can freely use the function.

```
from q23mod import f
f()
```

It is also possible to use from to import ALL of the attributes of a module using an asterisk. We'll be able to use the functions with prefixing it with the module name:

```
from q23mod import f
f()
```

Consider the following lines of code:

```
import math as mth
from math import sqrt

print("start")
```

Given the above lines of code, which of the following will calculate the square root of 7?

Correct! ✓ mth.sqrt(7)

Correct! ✓ 7**0.5

Correct! ✓ sqrt(7)

You Answered ✓ math.sqrt(7)

Question 5

1 / 1 pts

Classes are also namespaces. So far, we have been using objects (such as lists) as their methods without much regard to what is happening in the background. We'll look more at classes in the upcoming sections. Notice, however, that we can use the sort() function with a list, but not alone:

```
pets = ['dog', 'cat', 'fish', 'hamster']
pets.sort()
sort()
```

Consider the following lines of code:

```
def pop():
    print('corn')
pets = ['dog', 'cat', 'fish', 'hamster']
```

Which functions can we call?

☐ sort()

☐ lst.pop()

Correct! ✓ pets.sort()

Correct! [✓] pets.pop()

Correct! [✓] pop()

Quiz Score: **4** out of 5