A module foo

In the subsequent questions, assume that foo.py contains the following lines:

```python
def bar(x):
    print('name in bar:', __name__)
    return x * 2

def baz(x):
    return x / 2

print('name is: ', __name__)
if __name__ == '__main__':
    print('in main')
else:
    print('not in main')

print('final line')
```
Q1: What will happen?

Assume `foo.py` contains:

```python
def bar(x):
    print('name in bar:', __name__)
    return x * 2

def baz(x):
    return x / 2

print('name is: ', __name__)
if __name__ == '__main__':
    print('in main')
else:
    print('not in main')
print('final line')
```

What will be printed if we open `foo.py` in Spyder’s editor and click the green arrow run button?

<table>
<thead>
<tr>
<th>A</th>
<th>Nothing</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>name in bar: foo name is: foo not in main final line</td>
</tr>
<tr>
<td>C</td>
<td>name in bar: bar name is: <strong>main</strong> in main</td>
</tr>
<tr>
<td>D</td>
<td>name is: <strong>main</strong> in main final line</td>
</tr>
<tr>
<td>E</td>
<td>None of the above</td>
</tr>
</tbody>
</table>
Q2: What will happen?

Assume `foo.py` contains:

```python
def bar(x):
    print('name in bar: ', __name__)
    return x * 2

def baz(x):
    return x / 2

print('name is: ', __name__)
if __name__ == '__main__':
    print('in main')
else:
    print('not in main')
print('final line')
```

What will be printed if we type `import foo` into the Python console?

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Nothing</td>
</tr>
</tbody>
</table>
| B | name is: foo
   | not in main
   | final line            |
| C | name is: __main__
   | in main
   | final line            |
| D | name in bar: foo
   | name is: __main__
   | in main
   | final line            |
| E | None of the above      |
Q3: What will happen?

Assume `foo.py` contains:

```python
def bar(x):
    print('name in bar:', __name__)
    return x * 2

def baz(x):
    return x / 2

print('name is: ', __name__)
if __name__ == '__main__':
    print('in main')
else:
    print('not in main')

print('final line')
```

What will be printed if we type `import foo as spam` into the Python console? (first time)

<table>
<thead>
<tr>
<th></th>
<th>Nothing</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>name is: foo</td>
</tr>
<tr>
<td></td>
<td>not in main</td>
</tr>
<tr>
<td></td>
<td>final line</td>
</tr>
<tr>
<td>B</td>
<td>name is: spam</td>
</tr>
<tr>
<td></td>
<td>not in main</td>
</tr>
<tr>
<td></td>
<td>final line</td>
</tr>
<tr>
<td>C</td>
<td>name is: <strong>main</strong></td>
</tr>
<tr>
<td></td>
<td>in main</td>
</tr>
<tr>
<td></td>
<td>final line</td>
</tr>
<tr>
<td>D</td>
<td>None of the above</td>
</tr>
</tbody>
</table>

5
Q4: What will happen?

Assume `foo.py` contains:

```python
def bar(x):
    print('name in bar: ', __name__)
    return x * 2
def baz(x):
    return x / 2
print('name is: ', __name__)
if __name__ == '__main__':
    print('in main')
else:
    print('not in main')
print('final line')
```

What will be printed if we type `from foo import bar` into a the Python console? (first time)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Nothing</td>
</tr>
<tr>
<td>B</td>
<td>name in bar: foo</td>
</tr>
<tr>
<td>C</td>
<td>name is: foo not in main final line</td>
</tr>
<tr>
<td>D</td>
<td>name in bar: foo name is: foo not in main final line</td>
</tr>
<tr>
<td>E</td>
<td>None of the above</td>
</tr>
</tbody>
</table>
Q5: What will happen?

Assume `foo.py` contains:

```python
def bar(x):
    print('name in bar:', __name__)
    return x * 2

def baz(x):
    return x / 2

print('name is: ', __name__)
if __name__ == '__main__':
    print('in main')
else:
    print('not in main')
print('final line')
```

Assume we have just typed `import foo` into the console. What will be displayed if we then type `bar(2.0)`?

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>An error</td>
</tr>
<tr>
<td>B</td>
<td>name in bar: foo 4.0</td>
</tr>
<tr>
<td>C</td>
<td>name in bar: <strong>main</strong> 4.0</td>
</tr>
<tr>
<td>D</td>
<td>name in bar: foo name is: foo not in main final line 4.0</td>
</tr>
<tr>
<td>E</td>
<td>None of the above</td>
</tr>
</tbody>
</table>
Q6: What will happen?

Assume `foo.py` contains:

```python
def bar(x):
    print('name in bar:', __name__)
    return x * 2

def baz(x):
    return x / 2
print('name is: ', __name__)
if __name__ == '__main__':
    print('in main')
else:
    print('not in main')
print('final line')
```

Assume we have just typed `import foo` into the console. What will be displayed if we then type `foo.bar(2.0)`?

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>An error</td>
</tr>
<tr>
<td>B</td>
<td>name in bar: foo 4.0</td>
</tr>
<tr>
<td>C</td>
<td>name in bar: <strong>main</strong> 4.0</td>
</tr>
<tr>
<td>D</td>
<td>name in bar: foo name is: foo not in main final line 4.0</td>
</tr>
<tr>
<td>E</td>
<td>None of the above</td>
</tr>
</tbody>
</table>
Q7: What will happen?

Assume `foo.py` contains:

```python
def bar(x):
    print('name in bar: ', __name__)
    return x * 2

def baz(x):
    return x / 2

print('name is: ', __name__)
if __name__ == '__main__':
    print('in main')
else:
    print('not in main')
print('final line')
```

Assume we open `foo.py` in Spyder’s editor and clicked the green arrow run button. (new console)

What will be displayed when we type `foo.bar(2.0)`?

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>An error</td>
</tr>
<tr>
<td>B</td>
<td>name in bar: foo 4.0</td>
</tr>
<tr>
<td>C</td>
<td>name in bar: <strong>main</strong> 4.0</td>
</tr>
<tr>
<td>D</td>
<td>name in bar: foo name is: foo not in main final line 4.0</td>
</tr>
<tr>
<td>E</td>
<td>None of the above</td>
</tr>
</tbody>
</table>

9
Q8: What will happen?

Assume `foo.py` contains:

```python
def bar(x):
    print('name in bar:', __name__)
    return x * 2

def baz(x):
    return x / 2

print('name is:', __name__)
if __name__ == '__main__':
    print('in main')
else:
    print('not in main')
print('final line')
```

Assume we open `foo.py` in Spyder’s editor and clicked the green arrow run button. (new console)

What will be displayed when we type `bar(2.0)`?

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>An error</td>
</tr>
<tr>
<td>B</td>
<td>name in bar: foo 4.0</td>
</tr>
<tr>
<td>C</td>
<td>name in bar: <strong>main</strong> 4.0</td>
</tr>
<tr>
<td>D</td>
<td>name in bar: foo name is: foo not in main final line 4.0</td>
</tr>
<tr>
<td>E</td>
<td>None of the above</td>
</tr>
</tbody>
</table>
Assume `foo.py` contains:

```python
def bar(x):
    print('name in bar: ', __name__)
    return x * 2

def baz(x):
    return x / 2

print('name is: ', __name__)

if __name__ == '__main__':
    print('in main')
else:
    print('not in main')

print('final line')
```

Assume we have already typed `from foo import bar as spam` into the Python console.

What will be displayed when we type `spam(2.0)`?

<table>
<thead>
<tr>
<th></th>
<th>An error</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>name in bar: spam 4.0</td>
</tr>
<tr>
<td>B</td>
<td>name in bar: foo 4.0</td>
</tr>
<tr>
<td>C</td>
<td>name in bar: foo not in main final line 4.0</td>
</tr>
<tr>
<td>D</td>
<td>None of the above</td>
</tr>
</tbody>
</table>

**Q9: What will happen?**