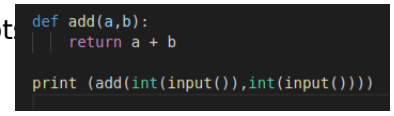
socrative

Week 1 Quiz

12 Questions

1. In the code pictured here, what will happen if the user attempts to input the values "h" and "p"?



134/151 **A** stop execution and p int a traceback

14/151 **B** p int "hp"

1/151 **C** p int 0

2/151 **D** p int "h + p"

2. A t y/except statement is used to:

2/116 **A** fix programming e rors

22/116 **B** prevent programming e rors from occu ing

92/116 **C** catch an e ror du ing program execution

0/116 **D** attempt to iterate over a list

3. In a t y/except block, code inside the finally statement will execute under which condition?

24/116 **A** after t y, but only when an exception is raised

17/116 **B** after t y, but only when an exception does not occur

73/116 **C** after t y, regardless of whether or not an exception is raised

2/116 **D** only if t y statement is not executed

4. In a t y/except block, code inside the else statement will execute under which condition?

46/120 **A** after t y, but only when an exception is raised

62/120 **B** after t y, but only when an exception does not occur

5/120 **C** after t y, regardless of whether or not an exception is raised

7/120 **D** only if t y statement is not executed

5. In a t y/except block, code inside the except statement will execute under which condition?

106/119 **A** after t y, but only when an exception is raised

1/119 **B** after t y, but only when an exception does not occur

2/119 **C** after t y, regardless of whether or not an exception is raised

10/119 **D** only if t y statement is not executed

6. The pathlib module allows you to add which type of functionality to your program?

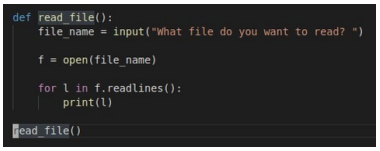
88/120 **A** interact with the host filesystem

0/120 **B** handle exceptions

2/120 **C** recursion

30/120 **D** all of the above

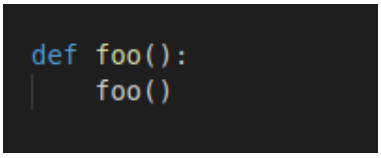
7. In a few words, desc ibe what is missing from this code sample.



```
def read_file():
    file_name = input("What file do you want to read? ")

    f = open(file_name)

    for l in f.readlines():
        print(l)

read_file()
```

i The file was opened, but not closed, so missing "f.close()"

8. The code pictured here is an example of what programming concept?



```
def foo():
    foo()
```

1/120 **A** abstraction

116/120 **B** recursion

2/120 **C** inheritance

1/120 **D** composition

9. Why is recursion impo tant for traversing tree like st uctures such as file systems?

3/120 **A** It reduces code complexity

2/120 **B** It suppo ts the infinite expandability of tree like st uctures

4/120 **C** It *ca n* be more efficient than loops

30/120 **D** A and B only

81/120 **E** All of the above

10. Which of the following values will be output from the code pictured here?

```
def recursive_sum(nested_list) -> int:
    total = 0

    for obj in nested_list:
        if type(obj) == list:
            total += recursive_sum(obj)

    return total

print(recursive_sum([1,2,[3,4],[5],6]))
```

66/120 **A** 0

36/120 **B** 21

17/120 **C** 11

1/120 **D** 1

i The value for total is never changed, so the final output will be 0

11. Which of the following code snippets will fix the previous function?

```
def recursive_sum(nested_list) -> int:
    total = 0

    for obj in nested_list:
        if type(obj) == list:
            total += recursive_sum(obj)

    return total

print(recursive_sum([1,2,[3,4],[5],6]))
```

111/121 **A**

```
for obj in nested_list:
    if type(obj) == list:
        total += recursive_sum(obj)
    else:
        total += obj
```

2/121 **B**

```
for obj in nested_list:
    if type(obj) == list:
        total += recursive_sum(obj)

    total += obj
```

8/121 **C**

```
for obj in nested_list:
    if type(obj) == list:
        total += recursive_sum(obj)
    else:
        total += 1
```

i obj is the value we are attempting to sum, so when obj is not of type list, it must be an integer and therefore should be added to the total.