Announcements & Logistics

- CS134 Scheduled Final: **Friday, May 17, 9:30 AM**
  - Room: **TCL 123**
- CS134 **Review Session** before Final:
  - Wednesday May 15, **4.30-5.30 pm**
  - Room: Wege (TCL 123)
- Bill Help hours on Thurs May 16: **2.30 - 4.30 pm (TCL 217)**
- **Practice Finals** are posted, along with solutions
  - Attempt thoroughly before checking solution key
  - If you have questions, bring it review session

Do You Have Any Questions?

CS134 (Review) : Jeopardy

Rules of the Game

- The team in control of the board chooses a category and point value
    - Higher-point-value questions are more challenging
- ALL teams start working on the solution and when a team is done, a team member raises a hand holding their written solution
- I will begin counting and other teams may raise their solution before the count reaches '5'
    - All answers must be written down on a piece of paper
    - Once a solution is raised, it is final!
- All teams that answered correctly earn points
- The first team to raise their hand that had a correct answer gets control of the board
- All teams that answered incorrectly lose points

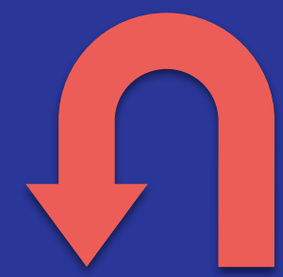
Game Board

Short & Sweet	Predict the Output	OOP	Loops and Recursion	Potpourri
2	2	2	2	2
3	3	3	3	3
5	5	5	5	5
7	7	7	7	7

Short & Sweet for 2 Points

This Python type is most appropriate to store unordered values but it does not store duplicates.

What is ....?



Short & Sweet for 3 Points

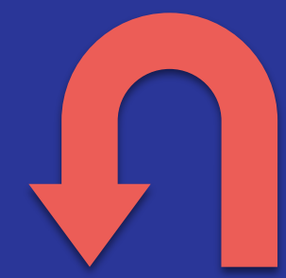
This expression from below DOES NOT give a TypeError.

A. `{1: 'o'} + {2: 'h'}`

B. `len(77777)`

C. `3 in range(10)`

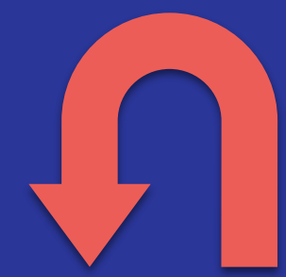
What is ....?



Short & Sweet for 5 Points

This is a one-line Python expression that converts 'a,b,c,d,e,f' to 'abcdef'.

What is ....?

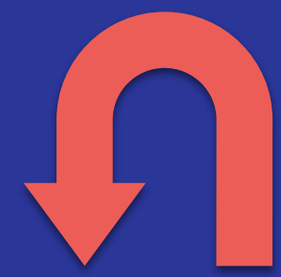


Short & Sweet for 7 Points

Given a list L of single-character digit strings, this is a one-line expression whose value is the integer that corresponds to concatenating the digits in reverse order, e.g.,

- if L is the list ['3', '4', '5'], the code should compute 543
- if L is the list ['5', '3', '7', '2'], the code should compute 2735

What is ....?

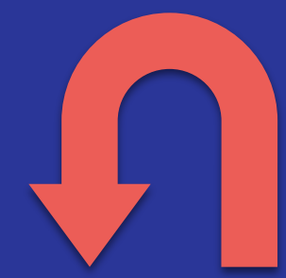


Predict the Output for 2 Points

This is the output printed by the following code:

```
print(print("hello"))
```

What is ....?

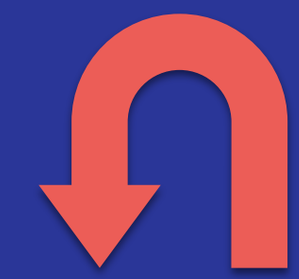


Predict the Output for 3 Points

This is the output printed by the following code:

```
x, y = 3, 8
def f():
    x, y = 6, 7
f()
print(x, y)
```

What is ....?

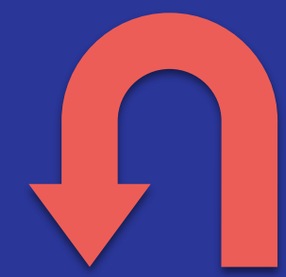


Predict the Output for 5 Points

This is the output printed by the following code:

```
t = ['5', '12', '3', '007']
print(sorted(t, key=int))
```

What is ....?

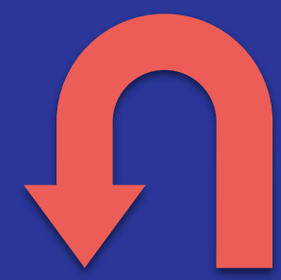


Predict the Output for 7 Points

This is the output printed by the following code:

```
d = {1: {2: 3}, 4: {5: 6}}
s = 0
for k1 in d:
    for k2 in d[k1]:
        s += k1 + d[k1][k2]
print(s)
```

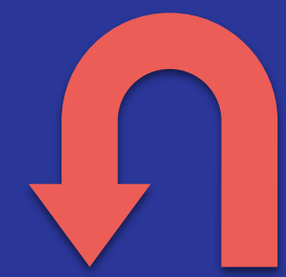
What is ....?



OOP for 2 Points

This is the special method called when an instance of a class is created.

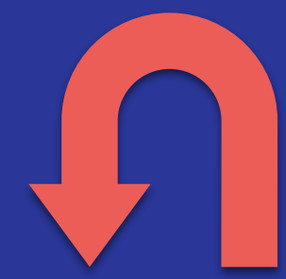
What is ....?



OOP for 3 Points

This is the special expression that is used instead of self when invoking a method of a parent class.

What is ....?

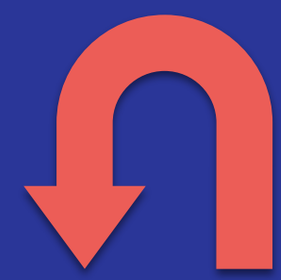


OOP for 5 Points

This is the attribute of Sample class that is not inherited by any of its subclass.

```
class Sample:
    def __init__(self, val1, val2, val3):
        self.a = val1
        self._b = val2
        self.__c = val3
```

What is ....?



OOP for 7 Points

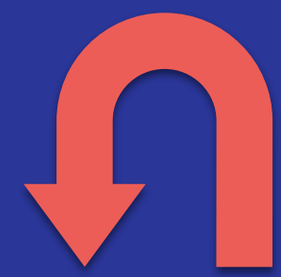
```
This is printed when the following code is run:

class Test:
   def __init__(self):
      print(self)

   def __str__(self):
      return "hello"

print(Test())
```

What is ....?

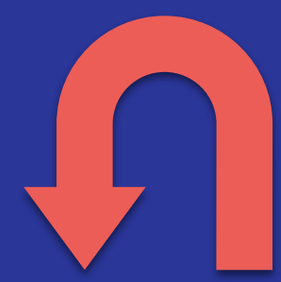


Loops and Recursion for 2 Points

This is the Big O Complexity of the following recursive function.

```
def halves(n):
    if n > 0:
      print(n)
      halves(n//2)
```

What is ....?

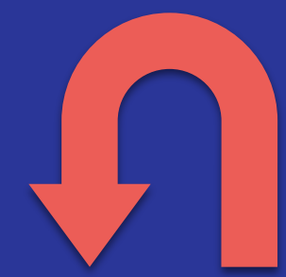


Loops and Recursion for 3 Points

```
This is printed when we run:


for i in range(2):
    for j in range(i):
        print(i, j)
```

What is ….?



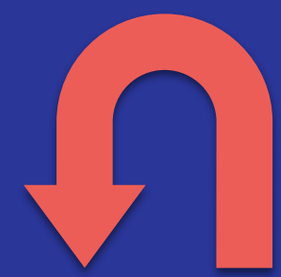
Loops and Recursion for 5 Points

This shape is drawn by the following recursion:

```
def draw(len, sides):
    if sides > 0:
        fd(len); lt(90)
        draw(len, sides-1)

draw(10, 4)
```

What is ....?

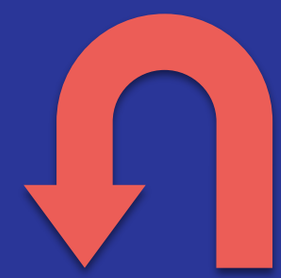


Loops and Recursion for 7 Points

What is the iterative function that is equivalent to this recursive function:

```
def mystery(num_lst):
    '''Assume num_lst is a list of numbers'''
    if len(num_lst) < 1:
        return 0
    else:
        return num_lst[0] + mystery(num_lst[1:])
```

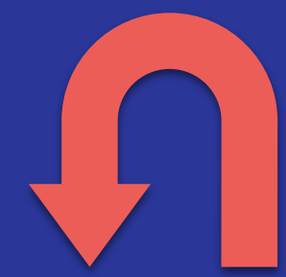
What is ....?



Potpourri for 2 Points

This is the name of the criterion for a data type to be a key in a dictionary

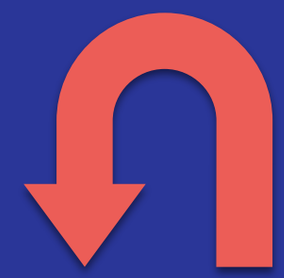
What is ....?



Potpourri for 3 Points

This is the Big O time complexity of an algorithm that compares each number in the list of numbers to every other number in the list (using a nested for loop) to determine if any pair adds up to a given target value.

What is ....?



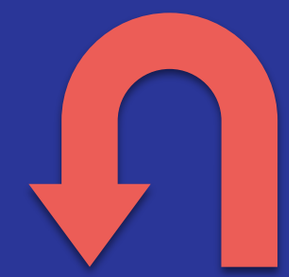
Potpourri for 5 Points

This is printed by the following code:

```
def optional(word, num=3):
    return word * num

if __name__ == "__main__":
    print(optional("a") + optional("z", 2))
```

What is ....?



Potpourri for 7 Points

This is the value of nums after this code is run:

```
nums = [1, 2, 3]
new = nums
new = new.append(4)
nums.append(new)
```

What is ....?