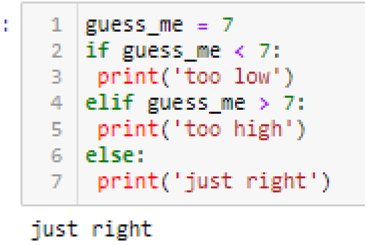
Name: Md Sajid Ansari
Full Stack Data Science- iNeuron 2021
Python Basic Assignment: Assignment 17

Question 1.

Assign the value 7 to the variable guess_me. Then, write the conditional tests (if, else, and elif) to print the string 'too low' if guess_me is less than 7, 'too high' if greater than 7, and 'just right' if equal to 7.

Answer 1:



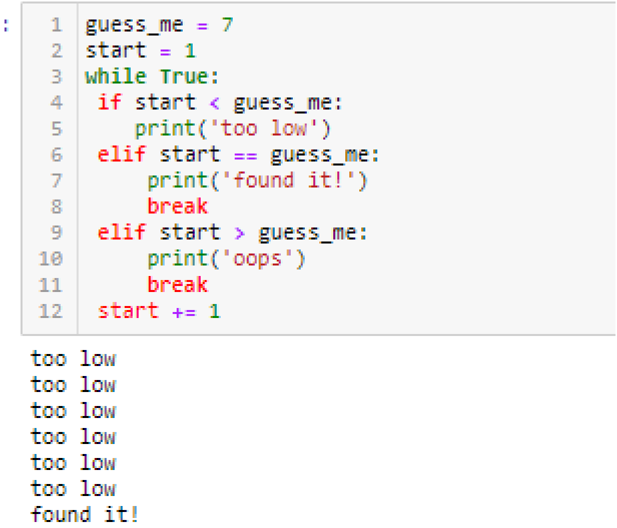
```
guess_me = 7
if guess_me < 7:
 print('too low')
elif guess_me > 7:
 print('too high')
else:
 print('just right')
```

```
just right
```

Question 2.

Assign the value 7 to the variable guess_me and the value 1 to the variable start. Write a while loop that compares start with guess_me. Print too low if start is less than guess me. If start equals guess_me, print 'found it!' and exit the loop. If start is greater than guess_me, print 'oops' and exit the loop. Increment start at the end of the loop.

Answer 2:



```
guess_me = 7
start = 1
while True:
 if start < guess_me:
    print('too low')
 elif start == guess_me:
     print('found it!')
     break
 elif start > guess_me:
     print('oops')
     break
 start += 1
```

```
too low
too low
too low
too low
too low
too low
found it!
```

```
guess_me = 7
start = 1
while True:
 if start < guess_me:
   print('too low')
 elif start == guess_me:
    print('found it!')
    break
 elif start > guess_me:
    print('oops')
    break
 start += 1
```

Question 3.

Print the following values of the list [3, 2, 1, 0] using a for loop.

Answer 3:

```
lst = [3,2,1,0]

for value in lst:

    print(value)
```

```
Output: 3
        2
        1
        0
```

Question 4.

Use a list comprehension to make a list of the even numbers in range(10)

Answer 4:

```
even = [number for number in range(10) if number % 2 == 0]
even
```

```
output: [0, 2, 4, 6, 8]
```

Question 5.

Use a dictionary comprehension to create the dictionary squares. Use range(10) to return the keys, and use the square of each key as its value.

Answer 5:

```
squares = {key: key*key for key in range(10)}
squares
```

```
output: {0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81}
```

Question 6.

Construct the set odd from the odd numbers in the range using a set comprehension (10).

Answer 6:

```
odd = {number for number in range(10) if number % 2 == 1}
odd
```

```
Output: {1, 3, 5, 7, 9}
```

Question 7.

Use a generator comprehension to return the string 'Got ' and a number for the numbers in range(10). Iterate through this by using a for loop.

Answer 7:

```
for thing in ('Got %s' % number for number in range(10)):
   print(thing)
```

```
Output: Got 0
Got 1
Got 2
Got 3
Got 4
Got 5
Got 6
Got 7
Got 8
Got 9
```

Question 8.

Define a function called good that returns the list ['Harry', 'Ron', 'Hermione'].

Answer 8:

```
def good():
   return ['Harry', 'Ron', 'Hermione']
good()
```

Output: ['Harry', 'Ron', 'Hermione']

Question 9.

Define a generator function called get_odds that returns the odd numbers from range(10). Use a for loop to find and print the third value returned.

Answer 9:

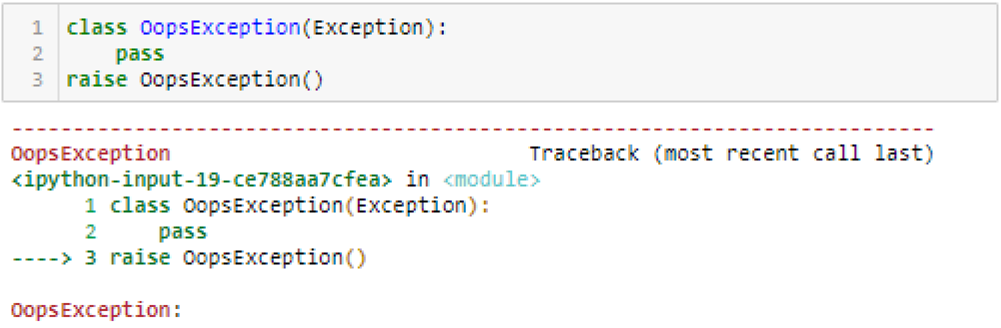
```
def get_odds():
   for number in range(1, 10, 2):
      yield number
count = 1
for number in get_odds():
   if count == 3:
      print("The third odd number is", number)
      break
   count += 1
```

output: The third odd number is 5

Question 10.

Define an exception called OopsException. Raise this exception to see what happens. Then write the code to catch this exception and print 'Caught an oops'.

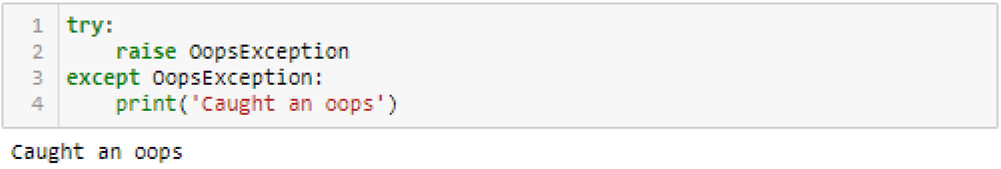
Answer 10:



```
class OopsException(Exception):
    pass
raise OopsException()
```

```
---------------------------------------------------------------------------
OopsException                             Traceback (most recent call last)
<ipython-input-19-ce788aa7cfea> in <module>
      1 class OopsException(Exception):
      2     pass
----> 3 raise OopsException()

OopsException: 
```



```
try:
    raise OopsException
except OopsException:
    print('Caught an oops')
```

```
Caught an oops
```

Question 11.

Use zip() to make a dictionary called movies that pairs these lists: titles = ['Creature of Habit', 'Crewel Fate'] and plots = ['A nun turns into a monster', 'A haunted yarn shop'].

Answer 11:

```
titles = ['Creature of Habit', 'Crewel Fate']
plots = ['A nun turns into a monster', 'A haunted yarn shop']
movies = dict(zip(titles, plots))
movies
```

```
Output: {'Creature of Habit': 'A nun turns into a monster',
 'Crewel Fate': 'A haunted yarn shop'}
```