Assignment 17 Solutions

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.

ANS:

In [1]:

```
def guess_me(guess_me):
    if guess_me < 7:
        print('too Low')
    elif guess_me > 7:
        print('too High')
    else:
        print('just Right')

guess_me(guess_me=7)
```

```
just Right
```

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.

ANS:

In [2]:

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

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

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

ANS:

In [4]:

```
in_list = [3,2,1,0]
for ele in in_list:
    print(ele)
```

```
3
2
1
0
```

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

ANS:

In [5]:

```
print([x for x in range(10+1) if x%2==0 ])
```

```
[0, 2, 4, 6, 8, 10]
```

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.

ANS:

In [7]:

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

Out[7]:

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

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

ANS:

In [8]:

```python
print({x for x in range(10) if x%2 !=0})
```

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

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

ANS:

In [9]:

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

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

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

ANS:

In [10]:

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

```
['Harry', 'Ron', 'Hermione']
```

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.

AND:

In [11]:

```
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
```

```
The third odd number is 5
```

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'.

ANS:

In [12]:

```
class OopsException(Exception):
    pass

def test(input):
    if input <0:
        raise OopsException()
try:
    test(-100)
except Exception as e:
    print('Caught in Oops ->',e)
```

```
Caught in Oops ->
```

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'].

ANS

In [14]:

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

Out[14]:

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