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:

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

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:

```
guess_me = 7
start = 1
while True:
   if start < guess_me:
      print('too low')
   elif start == guess_me:
      print('equal')
   start += 1
```

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

Answer:

```
guess_me =  [3, 2, 1, 0]
for p in guess_me:
   print(p, end=',')
```

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

Answer:

```
guess_me =  [ f for f in range(1,11) if f%2==0 ]
guess_me

OUTPUT >> [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.

Answer:

```
squares = {}
keys,values=[],[]
for k in range(10):
   keys.append(k)
   values.append(k*k)


for k,v in zip(keys,values):
   squares.update({k:v})

squares
```

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

Answer:

```
odd=set(f for f in range(0,10) if f%2 != 0)
odd

OUTPUT >>> {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.

Answer:

```
def genFun(num):
   for p in num:
      print(p,end=',')

genFun(*[range(0,10,1)])


OUTPUT >>> 0,1,2,3,4,5,6,7,8,9,
```

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

Answer:

```
def retList():
   return list(['Harry','Ron','Hermoine'])

retList()

OUTPUT >>> ['Harry', 'Ron', 'Hermoine']
```

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:

```
def get_odds(n):
   oddList=[f for f in n if f%2 != 0]
   print('the odd list is', oddList)
   print("the third element is", oddList[2])
```

```
get_odds([1,3,4,1,3,5,5,8,33,2,6,4,2])



OUTPUT >>> the odd list is [1, 3, 1, 3, 5, 5, 33]
                    the third element is 1
```

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:

```
try:
  fh = open("testfile", "w")
  fh.write("This is my test file for exception handling!!")
except OopsException:
  print("Caught an oops")
else:
  print("Written content in the file successfully")
  fh.close()


OUTPUT >>> Written content in the file successfully
```

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:

```
titles = ['Creature of Habit', 'Crewel Fate']
plots = ['A nun turns into a monster', 'A haunted yarn shop']
dict1={}
for key,value in zip(titles,plots):
   dict1.update({key:value})

dict1
```

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