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.

guess\_me = 7  
if guess\_me < 7:  
 print('too slow')  
elif guess\_me > 7:  
 print('too high')  
else:  
 print('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.

guess\_me = 7  
start = 1  
while start < guess\_me:  
 if start == guess\_me:  
 print('found it')  
 elif start < guess\_me:  
 print('too low')  
 elif start > guess\_me:  
 print('oops')  
 break;  
 start = start + 1

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

For i in list:

print(i)

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

list\_even = (i for i in range(20) if i % 2 == 0)  
for i in list\_even:  
 print(i)

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.

dict\_var = {1: 2, 2: 3, 3: 4, 4: 5, 5: 6, 6: 7, 7: 8, 8: 9, 9: 10}  
for i in range(1, 10):  
 print(dict\_var[i] \* dict\_var[i])

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

set\_list = set((i for i in range(20) if i % 2 == 1))  
for i in set\_list:  
 print(i)  
  
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.

listStr = ('Got ' + str(a) for a in range(10))  
for i in listStr:  
 print(i)

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

Def Good():   
 return ['Harry', 'Ron', 'Hermione']  
  
  
print(Good())

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.

def get\_odds():  
 return (i for i in range(10) if i % 2 == 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'.

class OopsException(RuntimeError):  
 def \_\_int\_\_(self):  
 self = self  
  
try:  
 raise OopsException()  
except OopsException:  
 print('Caught an 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'].

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