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 : guess\_me = 7**

**if guess\_me < 7:**

**print('too low')**

**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.

**Ans : 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**

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

**Ans : for value in [3, 2, 1, 0]:**

**print(value)**

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

**Ans : even = [number for number in range(10) if number % 2 == 0]**

**even**

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

**squares**

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

**Ans : odd = {number for number in range(10) if number % 2 == 1}**

**odd**

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

**print(thing)**

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

**Ans : def good():**

**return ['Harry', 'Ron', 'Hermione']**

**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.

**Ans : def get\_odds():**

**for number in range(1, 10, 2):**

**yield number**

**for count, number in enumerate(get\_odds(), 1):**

**if count == 3:**

**print("The third odd number is", number)**

**break**

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 :>>> class OopsException(Exception):**

**pass**

**>>>raise OopsException()**

**Traceback (most recent call last):**

**File "<stdin>", line 1, in <module>**

**\_\_main\_\_.OopsException**

**>>>**

**>>> 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'].

**Ans : titles = ['Creature of Habit', 'Crewel Fate']**

**plots = ['A nun turns into a monster', 'A haunted yarn shop']**

**movies = dict(zip(titles, plots))**

**movies**