ASSIGNMENT - 17(PYTHON)

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.

```
# Assign the value 7 to the variable guess_me
guess_me = 7

# Write conditional tests to print the corresponding string
if guess_me < 7:
    print('too low')
elif guess_me > 7:
    print('too high')
else:
    print('just right')
```

Output will be: 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.

```
# Assign the value 7 to the variable guess_me
guess_me = 7

# Assign the value 1 to the variable start
start = 1

# Write a while loop that compares start with guess_me
while True:
    if start < guess_me:
        print('too low')
    elif start == guess_me:
        print('found it')
        break
    else:
        print('oops')
        break
    start += 1
```

Output will be:

```
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.

```
# Define the list
values = [3, 2, 1, 0]

# Use a for loop to print each value in the list
for value in values:
    print(value)
```

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

```
# Use a list comprehension to make a list of the even numbers in range(10)
even_numbers = [x for x in range(10) if x % 2 == 0]

print(even_numbers)
```

Output will be:

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

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.

```
# Use a dictionary comprehension to create the dictionary squares
squares = {x: x**2 for x in range(10)}

print(squares)
```

Output will be:

```
{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).

```
# Use a set comprehension to construct the set of odd numbers in range(10)
odd = {x for x in range(10) if x % 2 != 0}

print(odd)
```

Output will be:

```
{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.

```
# Use a generator comprehension to create the generator
got_generator = ('Got ' + str(x) for x in range(10))

# Iterate through the generator using a for loop and print each value
for item in got_generator:
    print(item)
```

Output will be:

```
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'].

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

# Call the function and print the result
print(good())
```

Output will be:

```
['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.

```
def get_odds():
    for num in range(10):
        if num % 2 != 0:
            yield num

# Use a for loop to find and print the third value returned by get_odds
count = 0
for odd in get_odds():
    count += 1
    if count == 3:
        print(odd)
        break
```

Output will be:

```
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'.

```
# Define the custom exception
class OopsException(Exception):
    pass

# Raise the exception to see what happens
try:
    raise OopsException("Something went wrong!")
except OopsException as e:
    print('Caught an oops:', e)
```

Output will be:

```
Caught an oops: Something went wrong!
```

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'].

```
# Define the lists
titles = ['Creature of Habit', 'Crewel Fate']
plots = ['A nun turns into a monster', 'A haunted yarn shop']

# Use zip to pair the lists and create a dictionary
movies = dict(zip(titles, plots))

# Print the resulting dictionary
print(movies)
```

Output will be:

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