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.

Answers:

guess\_me = 7

If guess\_me < 7:

Print(“too low”)

elif guess\_me > 7:

Print(“too high”)

else :

Print(“just right”)

>> 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.

Answers:

guess\_me = 7

start =1

while True:

if start > guess\_me:

print ('too low')

elif start == guess\_me:

print('found it')

else:

print ('oops')

break

>> oops

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

Answers:

for i in [3,2,1,0]:

print(i)

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

Answers:

number\_list = [x for x in range(1,10) if x%2 ==0]

print(number\_list)

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.

Answers:

square = {x:x\*\*2 for x in range(1,10)}

print(square)

>> {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).

Answers:

set\_number = {x for x in range(1,10) if x%2 !=0}

print(set\_number)

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

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.

Answers:

generator = (('Got',x) for x in range(10))

for i in generator:

print (i)

>>

('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'].

Answers:

def good():

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

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.

Answers:

def get\_odds():

a = (x for x in range(10) if x%2 !=0 )

c=[]

for i in a:

c.append(i)

print (c[2])

get\_odds()

>> 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'.

Answers:

def test(func):

def new\_function(\*args,\*\*kwargs):

print('start')

answer ='end'

return answer

return new\_function

@test

def func():

pass

func()

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'].

Answers:

titles = ['Creature of Habit', 'Crewel Fate']

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

movies ={v:k for v,k in zip(titles,plots)}

print(movies)

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