1. Write a program in Python to allow the error of syntax to be handled using exception handling.

HINT: Use SyntaxError

>>> while True print('Hello world')

File "<stdin>", line 1

while True print('Hello world')

^

SyntaxError: invalid syntax

2. Write a program in Python to allow the user to open a file by using the argv module. If the entered name is incorrect throw an exception and ask them to enter the name again. Make sure to use read only mode.

import sys

print ("Number of arguments:", len(sys.argv), "arguments")

print ("Argument List:", str(sys.argv))

$ python test.py arg1 arg2 arg3

Number of arguments: 4 arguments.

Argument List: ['test.py', 'arg1', 'arg2', 'arg3']

3. Write a program to handle an error if the user entered a number more than four digits it should return “The length is too short/long !!! Please provide only four digits”

4. Create a login page backend to ask users to enter the username and password. Make sure to ask for a Re-Type Password and if the password is incorrect give chance to enter it again but it should not be more than 3 times.

print('Enter correct username and password combo to continue')

count=0

password=Hytu76E

username=bank\_admin

while password!='Hytu76E' and username!='bank\_admin' and count<4:

username=input('Enter username: ') and password=input('Enter password: ')

if password=='Hytu76E' and username=='bank\_admin':

print('Access granted')

else:

print('Access denied. Try again.')

5. Go through the link provided below to understand finally and raise concept: https://www.programiz.com/python-programming/exception-handling

# import module sys to get the type of exception

import sys

randomList = ['a', 0, 2]

for entry in randomList:

try:

print("The entry is", entry)

r = 1/int(entry)

break

except:

print("Oops!", sys.exc\_info()[0], "occurred.")

print("Next entry.")

print()

print("The reciprocal of", entry, "is", r)

6. Read doc.txt file using Python File handling concept and return only the even length string from the file. Consider the content of doc.txt as given below:

Hello I am a file

Where you need to return the data string Which is of even length

Make sure you return the content in The same link as it is present.

TASK SIX GENERATORS, LIST COMPREHENSION AND DECORATORS

1. Write a program in Python to find out the character in a string which is uppercase using list comprehension.

Extract Upper Case Characters

# Using list comprehension + isupper()

res = [char for char in test\_str if char.isupper()]

# printing result

print("The uppercase characters in string are : " + str(res)

2. Write a program to construct a dictionary from the two lists containing the names of students and their corresponding subjects. The dictionary should map the students with their respective subjects. Let’s see how to do this using for loops and dictionary comprehension.

HINT - Use Zip function also

# initializing lists

test\_keys = ["Rash", "Kil", "Varsha"]

test\_values = [1, 4, 5]

# Printing original keys-value lists

print ("Original key list is : " + str(test\_keys))

print ("Original value list is : " + str(test\_values))

# using naive method

# to convert lists to dictionary

res = {}

for key in test\_keys:

for value in test\_values:

res[key] = value

test\_values.remove(value)

break

# Printing resultant dictionary

print ("Resultant dictionary is : " + str(res))

Sample input: students = ['Smit', 'Jaya', 'Rayyan'] subjects = ['CSE', 'Networking', 'Operating System']

Expected output: {‘Smit’ : ’CSE’ , ’Jaya’ : ’Networking’ , ’Rayyan’ : ’Operating System’}

3. Learn More about Yield, next and Generators

def my\_gen():

n = 1

print('This is printed first')

# Generator function contains yield statements

yield n

n += 1

print('This is printed second')

yield n

n += 1

print('This is printed at last')

yield n

4. Write a program in Python using generators to reverse the string.

Input String = “Consultadd Training”

5. Write an example on decorators.

def my\_decorator(func):

def wrapper():

print("Something is happening before the function is called.")

func()

print("Something is happening after the function is called.")

return wrapper

def say\_whee():

print("Whee!")

say\_whee = my\_decorator(say\_whee)