Introduction to Python I (Exercises 07) Exceptions

1) Write a script that prompts for an integer. During the input you will convert the entered string into an integer (int function). Trap any exceptions and print a message letting the user know there was an error. Try entering letters instead of an integer, what happens? Are you able to trap the exception?

```
try:
    myinteger = int(input("Please enter an integer: "))
except:
    print("Invalid integer value ... !")
else:
    print(myinteger)
```

2) Try the same type of code but inside a function (call the function inputint()). The function will receive a string parameter containing the prompt to be displayed. When you call the function the function will return an integer inputted from the console. The function never exits until you enter an integer and it is successfully returned.

```
def inputint(prompt):

    integerFlag = True

    while integerFlag:
        try:
            myinteger = int(input(prompt))
        except:
            integerFlag = True
        else:
            integerFlag = False

    return myinteger


getinteger = inputint("Please enter an integer: ")

print(getinteger)
```

3) Write a simple script that divides a number (integer) by zero. Can you trap the exception and print a suitable message?

```
myinteger = 10

try:
    result = myinteger / 0 # This will cause an exception
except:
    print("Division by zero is not allowed!")
```

4) Prompt the user for a filename that does not exists. Try opening the file. Catch the exception and give a suitable error. Prompt for the file again or ‘q’ to exit.

```
filename = input("Enter file to be opened: ")

while filename != 'q':

    try:
        filehandle = open(filename, 'r')
    except:
        print("File {} does not exist!".format(filename))
    else:
        break

    filename = input("Enter file to be opened: ")
```

5) Write a script that generates 25 random numbers between 1 and 100. Add the random numbers to a list. Show which numbers occur more than once.

```
import random

mylist = []

for i in range(25):
    mylist.append(random.randint(1,100))

# Calculating occurrence of each element dictionary to store occurrences
# This is just one of many ways of checking the duplicates. Note: there are
# better ways of doing this in python, but given the knowledge you have
# acquired so far, this will be a good way of obtaining the desired result.

occurdict = {}

for i in mylist:
    occurdict[i] = occurdict.get(i,0)+1


for key, value in occurdict.items():
    if value > 1:
        print(key)
```

```
import random

mylist = []

for i in range(25):
    mylist.append(random.randint(1,100))

# Another way of checking for duplicates by using count and another list
# (once again read the disclaimer mentioned above). Once you learn about
# sets you will have more tools to obtain similar results

duplist = []

for i in mylist:
    if mylist.count(i) > 1 and i not in duplist:
        duplist.append(i)

for i in duplist:
    print(i)
```

6) Write a script containing a function ‘roll_dices()’ . The function will receive a dictionary containing several key:value pairs. The key:value pairs contain a dice name and a roll value (roll values will be zero to start). The dice names will be of the form: “name_type” (diceone_06, dicetwo_20, etc.). Inside the function, assign a random number to each dice depending on its type. (Note: the type has to be an integer!).

```
import random

def roll_dices(dices):
    for name, value in dices.items():
        randomValue = random.randint(1,int(name[-2:]))
        dices[name] = randomValue

mydices = {'energy_10': 0, 'power_06': 0, 'health_10': 0, 'wealth_20': 0}


roll_dices(mydices)


for name, value in mydices.items():
    print ('{:15}{}'.format(name, value))
```