Aarya Arban

S11-07

Assignment No.14 – File and Exception Handling

File Handling:

```
f = open("demofile.txt", "r")
print(f.read())
f = open("D:\myfiles\welcome.txt", "r")
print(f.read())
demofile.txt
f = open("demofile.txt", "r")
print(f.read(5))
#Read one line of the file:
f = open("demofile.txt", "r")
print(f.readline())
#By calling readline() two times, you can read the two first lines:
f = open("demofile.txt", "r")
print(f.readline())
print(f.readline())
#Loop through the file line by line:
f = open("demofile.txt", "r")
for x in f:
print(x)
#Close the file when you are finish with it:
f = open("demofile.txt", "r")
print(f.readline())
f.close()
```

```
#Open the file "demofile2.txt" and append content to the file:
f = open("demofile2.txt", "a")
f.write("Now the file has more content!")
f.close()
#open and read the file after the appending:
f = open("demofile2.txt", "r")
print(f.read())
#Open the file "demofile3.txt" and overwrite the content:
f = open("demofile3.txt", "w")
f.write("Woops! I have deleted the content!")
f.close()
#open and read the file after the overwriting:
f = open("demofile3.txt", "r")
print(f.read())
#Create a file called "myfile.txt":
f = open("myfile.txt", "x")
#Create a file called "myfile.txt":
f = open("myfile.txt", "x")
import os
os.remove("demofile.txt")
import os
if os.path.exists("demofile.txt"):
os.remove("demofile.txt")
else:
print("The file does not exist")
#To delete an entire folder, use the os.rmdir() method:
```

```
import os
os.rmdir("myfolder")
```

Exception Handling:

```
#ZeroDivision exception.
Enter a:10
Enter b:0
---------------------------------------------------------------------------
ZeroDivisionError Traceback (most recent call last)
<ipython-input-1-a104c1c193bd> in <module>()
 1 a = int(input("Enter a:"))
 2 b = int(input("Enter b:"))
----> 3 c = a/b
 4 print("a/b = %d" %c)
 5
#ZeroDivisionError: division by zero
a = int(input("Enter a:"))
b = int(input("Enter b:"))
c = a/b
print("a/b = %d" %c)
#other code:
print("Hi I am other part of the program")
try:
#block of code
except Exception1:
#block of code
except Exception2:
```

```
#block of code
#other code
try:
a = int(input("Enter a:"))
b = int(input("Enter b:"))
c = a/b
except:
print("Can't divide with zero")
Enter a:10
Enter b:0
try:
#block of code
except Exception1:
#block of code
else:
#this code executes if no except block is executed
try:
a = int(input("Enter a:"))
b = int(input("Enter b:"))
c = a/b
print("a/b = %d"%c)
# Using Exception with except statement. If we print(Exception) it will return exception cla
except Exception:
print("can't divide by zero")
print(Exception)
else:
```

```
print("Hi I am else block")
Enter a:10
Enter b:0
can't divide by zero
try:
a = int(input("Enter a:"))
b = int(input("Enter b:"))
c = a/b;
print("a/b = %d"%c)
except:
print("can't divide by zero")
else:
print("Hi I am else block")
Enter a:10
Enter b:0
can't divide by zero
try:
a = int(input("Enter a:"))
b = int(input("Enter b:"))
c = a/b
print("a/b = %d"%c)
# Using exception object with the except statement
except Exception as e:
print("can't divide by zero")
print(e)
else:
```

```
print("Hi I am else block")
Enter a:10
Enter b:0
can't divide by zero
try:
#this will throw an exception if the file doesn't exist.
fileptr = open("file.txt","r")
except IOError:
print("File not found")
else:
print("The file opened successfully")
fileptr.close()
File not found
try:
#block of code
except (<Exception 1>,<Exception 2>,<Exception 3>,...<Exception n>)
#block of code
else:
#block of code
try:
a=10/0;
except(ArithmeticError, IOError):
print("Arithmetic Exception")
else:
print("Successfully Done")
Arithmetic Exception
```

```
try:
# block of code
# this may throw an exception
finally:
# block of code
# this will always be executed
try:
fileptr = open("file2.txt","r")
try:
fileptr.write("Hi I am good")
finally:
fileptr.close()
print("file closed")
except:
print("Error")
Error
try:
age = int(input("Enter the age:"))
if(age<18):
raise ValueError
else:
print("the age is valid")
except ValueError:
print("The age is not valid")
Enter the age:12
```

```
The age is not valid
#Raise the exception with message
try:
num = int(input("Enter a positive integer: "))
if(num <= 0):
# we can pass the message in the raise statement
raise ValueError("That is a negative number!")
except ValueError as e:
print(e)
Enter a positive integer: 12
try:
a = int(input("Enter a:"))
b = int(input("Enter b:"))
if b is 0:
raise ArithmeticError
else:
print("a/b = ",a/b)
except ArithmeticError:
print("The value of b can't be 0")
Enter a:10
Enter b:0
The value of b can't be 0
class ErrorInCode(Exception):
def __init__(self, data):
self.data = data
def __str__(self):
```

```
return repr(self.data)
try:
raise ErrorInCode(2000)
except ErrorInCode as ae:
print("Received error:", ae.data)
Received error: 2000
```