Module 1: Introduction to everything

Lab 3: Exception Handling and File Handling

Objective

- **Exception Handling**
  - try and except
- **File Handling**

Exception Handling

Exceptions are raised when the program is syntactically correct, but the code resulted in an error. This error does not stop the execution of the program, however, it changes the normal flow of the program.

try and except

The basic terminology and syntax used to handle errors in Python is the **try** and **except** statements. The code which can cause an exception to occue is put in the *try* block and the handling of the exception is the implemented in the *except* block of code. The syntax form is:

```
try:
   You do your operations here...
   ...
except ExceptionI:


   If there is ExceptionI, then execute this block.

except ExceptionII:

   If there is ExceptionII, then execute this block.
   ...

else:

   If there is no exception then execute this block.
```

We can also just check for any exception with just using except: To get a better understanding of all this lets check out an example: We will look at some code that opens and writes a file:

```python
try:
    x = 1 / 0
except ZeroDivisionError:
    print('divided by zero')
    print('executed when exception occurs')
else:
    print('executed only when exception does not occur')
finally:
    print('finally block, always executed')
```

```
divided by zero
executed when exception occurs
finally block, always executed
```

File Handling

File Handling: File handling is the process of applying various operations on the file such as create , update , delete etc. File Handling in python: Python provides a number of in-built operations to manipulate the files.

File I/O : Helps you read your files

- Python provides a `file` object to read text/binary files.
- This is similar to the `FileStream` object in other languages.
- Since a `file` is a resource, it must be closed after use. This can be done manually, or using a context manager (**`with`** statement)

```python
with open('myfile.txt', 'w') as f:
    f.write("This is my first file!\n")
    f.write("Second line!\n")
    f.write("Last line!\n")


# let's verify if it was really created.
# For that, let's find out which directory we're working from
import os
print(os.path.abspath(os.curdir))
```

```
C:\Users\aspdi\Downloads\Week-01-Git_&_Python_intro\notebooks
```

```python
# read the file we just created
with open('myfile.txt', 'r') as f:
    for line in f:
        print(line)
```

```
This is my first file!

Second line!
```

```
Last line!
```

Thank You !!!