**Name : Shawn Louis Batch : B Roll No : 31**

Experiment No: 9

|  |  |
| --- | --- |
| **Topic:** | Program To Demonstrate File Handling in Python. |
| **Prerequisite:** | Knowledge of some programming language like C, Java, basic file handling operations |
| **Mapping With COs:** | CSL405.4 |
| **Objective**: | Creating, reading, updating, and deleting files using Python functions. |
| **Outcome:** | Students will have the skills to handle various file operations using Python. |
|  |
| **Bloom’s Taxonomy :** | Apply |
| **Theory/ Steps/ Algorithm/ Procedure:** | File handling is an important part of any web application. Python has several functions for creating, reading, updating, and deleting files.  **File Handling**  The key function for working with files in Python is the open() function. The open() function takes two parameters; *filename*, and *mode*.  There are four different methods (modes) for opening a file: |
| "r" - Read - Default value. Opens a file for reading, error if the file does not exist  "a" - Append - Opens a file for appending, creates the file if it does not exist "w" - Write - Opens a file for writing, creates the file if it does not exist  "x" - Create - Creates the specified file, returns an error if the file exists |
| In addition you can specify if the file should be handled as binary or text mode |
| "t" - Text - Default value. Text mode  "b" - Binary - Binary mode (e.g. images) |
| Syntax  To open a file for reading it is enough to specify the name of the file: f = open("demofile.txt") |

|  |  |  |
| --- | --- | --- |
|  | The code above is the same as:  f = open("demofile.txt", "rt")  Because "r" for read, and "t" for text are the default values, you do not need to specify them. | |
| **Note:** Make sure the file exists, or else you will get an error. | |
| **Open a File on the Server**  Assume we have the following file, located in the same folder as Python: | |
| demofile.txt |  |
|  |
| Hello! Welcome to demofile.txt This file is for testing purposes. Good Luck!  To open the file, use the built-in open() function.  The open() function returns a file object, which has a read() method for read- ing the content of the file: | |
| Example: **Open the file:** | |
| f = open("demofile.txt", "r") print(f.read())  **Read Only Parts of the File**  By default the read() method returns the whole text, but you can also specify how many characters you want to return: | |
| Example: **Return the 5 first characters of the file:** | |
| f = open("demofile.txt", "r") print(f.read(**5**))  **Read Lines**  You can return one line by using the readline() method: | |
| Example: **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: |
| Example: **Read two lines of the file:** |
| f = open("demofile.txt", "r") print(f.readline()) print(f.readline()  By looping through the lines of the file, you can read the whole file, line by line: |
| Example: **Loop through the file line by line:** |
| f = open("demofile.txt", "r") for x in f:  print(x)  **Close Files**  It is a good practice to always close the file when you are done with it. |
| Example: **Close the file when you are finish with it:** |
| f = open("demofile.txt", "r") print(f.readline())  f.close()  **Note:** You should always close your files, in some cases, due to buffering, changes made to a file may not show until you close the file.  **Write to an Existing File**  To write to an existing file, you must add a parameter to the open() function: "a" - Append - will append to the end of the file  "w" - Write - will overwrite any existing content |
| Example: **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()) |

|  |  |
| --- | --- |
|  | Example: **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 appending:  f = open("demofile3.txt", "r") print(f.read())  **Create a New File**  To create a new file in Python, use the open() method, with one of the fol- lowing parameters:  "x" - Create - will create a file, returns an error if the file exist "a" - Append - will create a file if the specified file does not exist "w" - Write - will create a file if the specified file does not exist |
| Example  Create a file called "myfile.txt": |
| f = open("myfile.txt", "x")  Result: a new empty file is created! |
| Example: **Create a new file if it does not exist:** |
| f = open("myfile.txt", "w")  **Delete a File**  To delete a file, you must import the OS module, and run its os.remove() function: |
| Example:**Remove the file "demofile.txt":** |
| import os os.remove("demofile.txt")  **Check if File exist:**  To avoid getting an error, you might want to check if the file exists before you try to delete it: |
| Example: **Check if file exists, *then* delete it:** |

|  |  |
| --- | --- |
|  | import os  if os.path.exists("demofile.txt"): os.remove("demofile.txt")  else:  print("The file does not exist")  **Delete Folder**  To delete an entire folder, use the os.rmdir() method: |
| Example: **Remove the folder "myfolder":** |
| import os  os.rmdir("myfolder") |
| Note: You can only remove empty folders. |
|  |
| **Experiments:** | 1. Practice all the small Examples mentioned in the Theory/Steps/ Algorithm/ Procedure-Section of the same document. 2. Write a Python Program to open a file called File1.txt, and then read through the file line-by-line. Add few more lines to the file, rename the file “Newfile.txt” and then print the content of the file. 3. Write a Python Program to merge two files into a third file. |
| **Deliverables:** | 1. **Practice all the small Examples mentioned in the Theory/Steps/ Algorithm/ Procedure-Section of the same document.**     import os  fileName = "myfile.txt"  if os.path.isfile(fileName):  f = open(fileName)  print(f.read())  f.close()  else:  print(fileName + ' does not exist')    import os  fileName = "myfile.txt"  if os.path.isfile(fileName):  f = open(fileName)  print(f.read(5))  f.close()  else:  print(fileName + ' does not exist')    import os  fileName = "myfile.txt"  if os.path.isfile(fileName):  f = open(fileName)  print(f.readline())  f.close()  else:  print(fileName + ' does not exist')    import os  fileName = "myfile.txt"  if os.path.isfile(fileName):  f = open(fileName)  for x in f:  print(x)  f.close()  else:  print(fileName + ' does not exist')    import os  fileName = "myfile.txt"  if os.path.isfile(fileName):  f = open(fileName, 'a')  f.write('\n'+"Now the file has more content!")  f.close()  f = open(fileName)  print(f.read())  f.close()  else:  print(fileName + ' does not exist')    import os  fileName = "myfile.txt"  if os.path.isfile(fileName):  f = open(fileName, 'w')  f.write("Woops! I have deleted the content!")  f.close()  f = open(fileName)  print(f.read())  f.close()  else:  print(fileName + ' does not exist')    f = open("myfile1.txt", 'x')  OR f = open(“myfile1.txt”, ‘w’)    import os  fileName = "myfile.txt"  if os.path.isfile(fileName):  os.remove(fileName)  else:  print(fileName + ' does not exist')    import os  os.mkdir('TestDirectory')    import os  os.rmdir(‘TestDirectory’)     * **2. Write a Python Program to open a file called File1.txt, and then read through the file line-by-line. Add few more lines to the file, rename the file “Newfile.txt” and then print the content of the file.** * import os * fileName = "File1.txt" * if os.path.isfile(fileName): * f = open(fileName, 'r') * for x in f: * print(x.rstrip()) * f.close() * f = open(fileName, 'a') * n = int(input("\nEnter number of lines to be appended : ")) * for i in range(0, n): * string = input("Enter string to be appended : ") * f.write('\n' + string) * f.close() * newName = 'changed.txt' * os.rename(fileName, newName) * f = open(newName, 'r') * print() * print(f.read()) * f.close() * else: * print(fileName + ' does not exist')        * **3. Write a Python Program to merge two files into a third file.**   import os  file1 = "File1.txt"  file2 = "File2.txt"  file3 = "File3.txt"  if os.path.isfile(file1 and file2):  with open(file1) as f:  content1 = f.read()  print("File1 contains : ")  print(content1)  with open(file2) as f:  content2 = f.read()  print("\nFile2 contains : ")  print(content2)  with open(file3, 'w+') as f:  f.write(content1 + '\n' + content2)  f.seek(0)  print("\nFile3 is created and it contains : ")  print(f.read())    else:  print(fileName + ' does not exist') |
| **Conclusion:** | Thus we have successfully able to create, reading, update, and delete files  using Python functions. |
| **References:** | <https://www.w3schools.com/python/python_file_handling.asp> |

Don Bosco Institute of Technology Department of Computer Engineering

Academic year – 2019-20 Open Source Technology Lab

Assessment Rubric for Experiment No.: 9

Performance Date : Submission Date :

**Title of Experiment** : Program To Demonstrate File Handling in Python

**Year and Semester** : 2nd Year and IVth Semester

**Batch** : Computer

Name of Student : Shawn Louis

Roll No. : 31

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Performance** | **Poor** | **Satisfactory** | **Good** | **Excellent** | **Total** |
| 2 points | 3 points | 4 points | 5 points |  |
| **Results and Documentatio ns** | **Poor** | **Satisfactory** | **Good** | **Excellent** |
| 2 points | 3 points | 4 points | 5 points |
| **Timely Submission** | **Submissio n beyond 14 days of the deadline** | **Late submission till 14 days** | **Late submission till 7 days** | **Submission on time** |
| 2 points | 3 points | 4 points | 5 points |

Signature

(Sana Shaikh)