

Compiler Design(18CSC304J)

Experiment 1

FILE OPERATIONS

20/01/2021

Harsh Goel

RA1811003010185

Aim: To study and perform file handling operations like adding, deleting and displaying and updating data.

Language: Python 3.7

Procedure:

1. Create a file or select the file for performing the operations on.
2. Start any python IDE and type the necessary code.
3. Run the code and perform the operations required.
4. Note the output and document it.

Algorithm:

- **Adding data to file:**
 - Open the file,input the data to be added and write the data in it using write() function.close the file.
- **Deleting data from file:**
 - Open the file, input the data to be deleted, search the data and delete the data in it.close the file.
- **Displaying data of file:**
 - Open the file, print the content of file,close the file.
- **Updating data of file:**
 - Open the file, input the data to be updated, search the data and write the new data in it.close the file.

Code Snippet:

```
import os.path
file = input('Input file name with extension -> ')
ls = [1, 2, 2]
if not os.path.exists(file):
    with open(file, 'w') as fp:
        pass
else:
    print('FILE ALREADY EXISTS!')
string = ' '
op = '0'
while op != '6':
    print('\n')
    print('1. Add')
    print('2. Delete')
    print('3. Display')
    print('4. Calculate')
    print('5. Update')
    print('6. Exit')
    op = input('Choose one of the following operations -> ')

    if op == '1':
        ls = []
        print('Input the data elements : ')
        for i in range(4):
            x = input()
            ls.append(x)
        string = ' '
        with open(file, 'a') as fp:
            fp.write(string.join(ls) + '\n')

    if op == '2':
        with open(file, 'r+') as fp:
            lines = fp.readlines()
            idx = int(input('Enter the index of record you want to delete -
> '))
            del lines[idx]
            fp.seek(0)
            fp.truncate(0)
            for line in lines:
                fp.write(line)

    if op == '3':
        with open(file, 'r') as fp:
            lines = fp.readlines()
            print('\nData : \n')
            for line in lines:
                print(line)
```

```

if op == '4':
    with open(file, 'r') as fp:
        r = fp.readlines()
        ls = []
        idx = int(input('Enter the index of record for which you want to
o calculate the sum -> '))
        ls = r[idx].split(' ')
        ls[-1] = ls[-1].split('\n')[0]
        ls.pop(0)
        sum = 0
        for n in ls:
            sum = sum + int(n)
        print('\nTotal marks of student at index {} is {}'.format(idx,
sum))

if op == '5':
    with open(file, 'r+') as fp:
        lines = fp.readlines()
        idx = int(input('Enter the index of record you want to update -
> '))
        ls = []
        print('Input the data elements : ')
        for i in range(4):
            x = input()
            ls.append(x)
        ls.append('\n')
        upd = ' '.join(ls)
        for line in lines:
            ls = line.split(' ')
        lines[idx] = upd
        fp.seek(0)
        fp.truncate(0)
        for line in lines:
            fp.write(line)

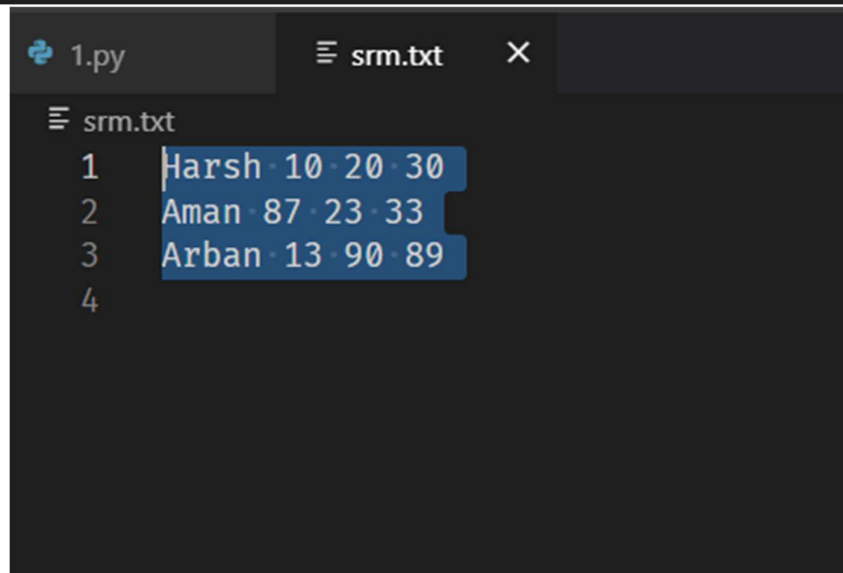
if op == '6':
    break

```

Code Snippet:

```
PS C:\Users\HARSH-PC\Desktop\college\COMPILER_DESIGN\exp_1> py 1.py
Input file name with extension -> srm.txt
```

```
1. Add
2. Delete
3. Display
4. Calculate
5. Update
6. Exit
Choose one of the following operations -> 1
Input the data elements :
Harsh
10
20
30
```



```
1.py srm.txt X
srm.txt
1 Harsh 10 20 30
2 Aman 87 23 33
3 Arban 13 90 89
4
```

```
1. Add
2. Delete
3. Display
4. Calculate
5. Update
6. Exit
Choose one of the following operations -> 3
Data :
Harsh 10 20 30
```

```
1. Add
2. Delete
3. Display
4. Calculate
5. Update
6. Exit
Choose one of the following operations -> 4
Enter the index of record for which you want to calculate the sum -> 0

Total marks of student at index 0 is 60
```

Result:

The file operations were studied and successfully implemented using a python code and a file.