

Programming Assignment-8

Files and Exception handling

1. Write a Python function that takes two file names, datafile1 and datafile2 as input. The function should read the contents of the file datafile1 line by line and should write them to another file datafile2 after adding a newline at the end of each line
2. Write a Python function that reads a file file1 and displays the number of words and the number of vowels in the file
3. Write a Python function that takes data to be stored in the file 'file1' as interactive input from the user until he responds with nothing as input. Each line (or paragraph) taken as input from the user should be capitalized, and stored in the file 'file1'.
4. Write a Python function that reads the file file1 and copies only alternative lines to another file file2. Alternative lines copied should be the odd numbered lines.
5. What will be the output produced on executing function inverse1 when the following input is entered as the value of variable num:
(a)5 (b)0 (c)2.0 (d)x (e)None

def inverse1()

```
    try:

        num = input('Enter the number:')
        num = float(num)
        inverse = 1.0 / num

    except ValueError:

        print('ValueError')

    except TypeError:

        print('TypeError')

    except ZeroDivisionError:

        print('ZeroDivisionError')

    except:

        print('Any other Error')

    else:

        print(inverse)

    finally:

        print('Function inverse completed')
```

6. Examine the following function percentage:

def percentage(marks, total):

```

try:
    percent = (marks / total) * 100
except ValueError:
    print('ValueError')
except TypeError:
    print('TypeError')
except ZeroDivisionError:
    print('ZeroDivisionError')
except:
    print('Any other Error')
else:
    print(percent)
finally:
    print('Function percentage completed')

```

Determine the output for the following function calls:

- (a) `percentage(150.0, 200.0)`
- (b) `percentage(150.0, 0.0)`
- (c) `percentage('150.0', '200.0')`

9. Identify two exceptions that may be raised while executing the following statement:

```
result = a + b
```

10. What will be the output for the following code snippets if the file being opened does not exist:

```

(a). try:
    f = open('file1.txt', 'r')
except IOError:
    print('Problem with Input Output...\n')

```

```
else:
```

```
    print('No Problem with Input Output...')
```

11. Consider the following program. Check for the error (if any), otherwise write the output.

```
f=open('PYTHON','w')  
f.write('"i am great" and ')  
f.write('" failure is a part of success"')  
f=open('PYTHON','r')  
print(f.read())  
f.close()
```

12. Consider the following program. Check for the error (if any), otherwise write the output.

```
f=open('file1','r')  
f.write('""work is worship""')  
f.close()
```

13. Consider the following program. Write the output.

```
(a). f=open('PYTHON','w')  
f.write('failure is a part of success')  
f = open('PYTHON', 'r')  
print(f.read(4))  
f.close()  
(b). f=open('PYTHON','w')  
f.write('failure is a part of success')  
f = open('PYTHON', 'r')  
print(f.read())  
f.close()
```

14. Consider the following program. Write the output.

```
f=open('PYTHON','w')  
f.write('failure is a part of success also, i am fine')
```

```
f = open('PYTHON', 'r')  
print(f.readline())  
f.close()
```

15. Consider the following program. Write the output.

```
f = open('PYTHON', 'w')  
description=['we either choose the pain of discipline \n', 'or\n', 'the pain of regret\n']  
f.writelines(description)  
f.close()  
f = open('PYTHON', 'r')  
print(f.read())  
f.close()
```

16. Define a stack along with push() and pop() operations using a list

17. Define a queue along with insert() and delete() operations using a list