**MODULE – 4**

1. **What is File function in python? What is keywords to create and write file.**

Python file object **provides methods and attributes to access and manipulate files**. Using file objects, we can read or write any files. Whenever we open a file to perform any operations on it, Python returns a file object. To create a file object in Python use the built-in functions, such as open() and os.

1. 'w' – open a file for writing. If the file doesn't exist, the open() function creates a new file. Otherwise, it'll overwrite the contents of the existing file.
2. 'x' – open a file for exclusive creation. If the file exists, the open() function raises an error ( FileExistsError ).
3. **Write a Python program to read an entire text file.**

def file\_read(fname):

        txt = open(fname)

        print(txt.read())

file\_read('test.txt')

What is Python language?

Python is a widely used high-level, general-purpose, interpreted, dynamic programming language.Its decepts in fewer lines of code than possible inlanguages such as C++ or Java.

1. **Write a Python program to append text to a file and display the text.**

def file\_read(fname):

from itertools import islice

with open(fname, "w") as myfile:

          myfile.write("Python Exercise\n")

          myfile.write("Java Exercises")

        txt = open(fname)

        print(txt.read())

file\_read('test.txt')

1. **Write a Python program to read first n lines of a file**

def file\_read\_from\_head(fname, nlines):

        from itertools import islice

        with open(fname) as f:

                for line in islice(f, nlines):

                       print(line)

file\_read\_from\_head('test.txt',2)

What is Python language?

Python is a widely used high-level, general-purpose, interpreted, dynamic programming language.Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than possible in

1. **Write a Python program to read last n lines of a file.**

def read\_lastnlines(fname,n):

    with open('test.txt') as f:

        for line in (f.readlines()[-n:]):

            print(line)

read\_lastnlines('test.txt',2)

It features a dynamic type system and automatic memory management and has a large and comprehensive standard library.

familiar with Python.

1. **Write a Python program to read a file line by line and store it into a list.**

def file\_read(fname):

        with open(fname) as f:

               content\_list=f.readlines()

                print(content\_list)

file\_read('test.txt')

['What is Python language? \n', 'Python is a widely used high-level, general-purpose, interpreted\n', 'dynamic programming language.Its design philosophy emphasizes code readability\n', 'its syntax allows programmers to express concepts in fewer lines of code than possible in\n', 'languages such as C++ or Java. \n', 'It features a dynamic type system and automatic memory management and has a large and comprehensive standard library.\n', 'familiar with Python.']

1. **Write a Python program to read a file line by line store it into a variable.**

def file\_read(fname):

        with open (fname, "r") as myfile:

                data=myfile.readlines()

                print(data)

file\_read('test.txt')

['What is Python language? \n', 'Python is a widely used high-level, general-purpose, interpreted\n', 'dynamic programming language.Its design philosophy emphasizes code readability\n', 'its syntax allows programmers to express concepts in fewer lines of code than possible in\n', 'languages such as C++ or Java. \n', 'It features a dynamic type system and automatic memory management and has a large and comprehensive standard library.\n', 'familiar with Python.']

1. **Write a python program to find the longest words.**

def longest\_word(filename):

    with open(filename, 'r') as infile:

             words =infile.read().split()

    max\_len = len(max(words, key=len))

    return [word for word in words if len(word) == max\_len]

print(longest\_word('test.txt'))

['general-purpose,']

1. **Write a Python program to count the number of lines in a text file.**

def file\_lengthy(fname):

        with open(fname) as f:

                for i, l in enumerate(f):

                        pass

        return i + 1

print("Number of lines in the file: ",file\_lengthy("test.txt"))

Number of lines in the file: 7

**10) Write a Python program to count the frequency of words in a file.**

from collections import Counter

def word\_count(fname):

        with open(fname) as f:

                return Counter(f.read().split())

print("Number of words in the file :",word\_count("test.txt"))

**11)Write a Python program to write a list to a file.**

color = ['Red', 'Green', 'White','Black', 'Pink', 'Yellow']

with open('test.txt', "w") as myfile:

        for c in color:

                myfile.write("%s\n" % c)

content = open('test.txt')

print(content.read())

Red

Green

White

Black

Pink

Yellow

**12) Write a Python program to copy the contents of a file to another file.**

with open('first.txt','r') as firstfile, open('second.txt','a') as secondfile:

    for line in firstfile:

              secondfile.write(line)

**13) Explain Exception handling? What is an Error in Python?**

If you have some *suspicious* code that may raise an exception, you can defend your program by placing the suspicious code in a **try:** block. After the try: block, include an **except:** statement, followed by a block of code which handles the problem as elegantly as possible.

Errors are **the problems in a program due to which the program will stop the execution**. On the other hand, exceptions are raised when some internal events occur which changes the normal flow of the program. Two types of Error occurs in python. Syntax errors. Logical errors (Exceptions).

There are mainly three kinds of distinguishable errors in Python: **syntax errors, exceptions and logical errors.**

**14) How many except statements can a try-except block have? Name Some built-in exception classes:**

 A try statement can have more **than one** except clause.

|  |  |
| --- | --- |
| ZeroDivisionError | Raised when the second operand of division or modulo  operation is zero. |
| ValueError | Raised when a function gets an argument of correct type  but improper value. |
| SyntaxError | Raised by parser when syntax error is encountered. |
| IndentationError | Raised when there is incorrect indentation. |

**15) When will the else part of try-except-else be executed?**

The else part is executed when no exception occurs.

**16) Can one block of except statements handle multiple exception?**

**yes**, like except TypeError, SyntaxError [,…]

**17) When is the finally block executed?**

The finally block always executes **when the try block exits**. This ensures that the finally block is executed even if an unexpected exception occurs.

**18) What happens when „1‟== 1 is executed?**

 it simply evaluates to false and does not raise any exception.

**19) How Do You Handle Exceptions With Try/Except/Finally In Python? Explain with coding snippets.**

In Python, exceptions can be handled **using a try statement.** **The critical operation which can raise an exception is placed inside the try clause.** **The code that handles the exceptions is written in the except clause**. We can thus choose what operations to perform once we have caught the exception.

**20) Write python program that user to enter only odd numbers, else will raise an exception.**

**21) What are oops concepts? Is multiple inheritance supported in python.**

Object-oriented programming is a model that provides different types of concepts, such as **inheritance, abstraction, polymorphism**, etc. These concepts aim to implement real-world entities in programs. They create working methods and variables to reuse them without compromising security.

**A class can be derived from more than one base class in Python**, similar to C++. This is called multiple inheritance. In multiple inheritance, the features of all the base classes are inherited into the derived class.