Que: What is File function in python? What is keywords to create and write file.

Ans:

The file handling plays an important role when the data needs to be stored permanently into the file. A file is a named location on disk to store related information. We can access the stored information (non-volatile) after the program termination.

The file-handling implementation is slightly lengthy or complicated in the other programming language, but it is easier and shorter in Python.

In Python, files are treated in two modes as text or binary. The file may be in the text or binary format, and each line of a file is ended with the special character.

Hence, a file operation can be done in the following order.

- Open a file
- Read or write - Performing operation
- Close the file

Opening a file

Python provides an **open()** function that accepts two arguments, file name and access mode in which the file is accessed. The function returns a file object which can be used to perform various operations like reading, writing, etc.

Syntax:

file object = open(<file-name>, <access-mode>, <buffering>)

Writing the file

To write some text to a file, we need to open the file using the open method with one of the following access modes.

w: It will overwrite the file if any file exists. The file pointer is at the beginning of the file.

a: It will append the existing file. The file pointer is at the end of the file. It creates a new file if no file exists.

que:Write a Python program to read an entire text file.

Ans:

```
def file_read(fname):
        txt = open(fname)
        print(txt.read())
file_read('test.txt')
```

que:Write a Python program to append text to a file and display the text.

Ans:

```
def file_read(fname):
        from itertools import islice
        with open(fname, "w") as myfile:
                myfile.write("Python Exercises\n")
                myfile.write("Java Exercises")
        txt = open(fname)
        print(txt.read())
file_read('abc.txt')
```

que:Write a Python program to read first n lines of a file.

Ans:

```
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)
```

que:Write a Python program to read last n lines of a file.

Ans:

```
import sys
import os
def file_read_from_tail(fname,lines):
    bufsize = 8192
    fsize = os.stat(fname).st_size
    iter = 0
    with open(fname) as f:
        if bufsize > fsize:
            bufsize = fsize-1
            data = []
            while True:
                iter +=1
                f.seek(fsize-bufsize*iter)
                data.extend(f.readlines())
                if len(data) >= lines or f.tell() == 0:
                    print(''.join(data[-lines:]))
                    break

file_read_from_tail('test.txt',2)
```

que:Write a Python program to read a file line by line and store it into a list

Ans:

```
def file_read(fname):
    with open(fname) as f:
        #Content_list is the list that contains the read lines.
        content_list = f.readlines()
        print(content_list)

file_read(\'test.txt\')
```

que:Write a Python program to read a file line by line store it into a variable.

Ans:

```
def file_read(fname):
        with open (fname, "r") as myfile:
                data=myfile.readlines()
                print(data)
file_read('test.txt')
```

que:Write a python program to find the longest words.

Ans:

```
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'))
```

que:Write a Python program to count the number of lines in a text file.

Ans:

```
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")
```

que:Write a Python program to count the frequency of words in a file.

Ans:

```
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"))
```

que:Write a Python program to write a list to a file.

Ans:

```
color = ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']
with open('abc.txt', "w") as myfile:
    for c in color:
        myfile.write("%s\n" % c)


content = open('abc.txt')
```

```
print(content.read())
```

que:Write a Python program to copy the contents of a file to another file.

Ans:

```
from shutil import copyfile

copyfile('test.py', 'abc.py')
```

que:Explain Exception handling? What is an Error in Python?

Ans:

An exception can be defined as an unusual condition in a program resulting in the interruption in the flow of the program. Whenever an exception occurs, the program stops the execution, and thus the further code is not executed. Therefore, an exception is the run-time errors that are unable to handle to Python script.

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

Ans:

How many except statements can a try-except block have? Explanation: There has to be **at least one** except statement. 2

que:When will the else part of try-except-else be executed?

Ans:

The else part is executed when no exception occurs.

que:Can one block of except statements handle multiple exception?

Ans:

Each type of exception can be specified directly. There is no need to put it in a list.

que:When is the finally block executed?

Ans:

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

Que: What happens when „1‟== 1 is executed?

Ans:

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

Que: How Do You Handle Exceptions With Try/Except/Finally In Python? Explain with coding snippets.

Ans:

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.

Que: Write python program that user to enter only odd numbers, else will raise an exception.

Ans:

```
num = int(input("Enter a number: "))
mod = num % 2
if mod > 0:
   print("This is an odd number.")
else:
   print("This is an even number.")
```

Que: What are oops concepts? Is multiple inheritance supported in java

Ans:

Multiple Inheritance is a feature of an object-oriented concept, where a class can inherit properties of more than one parent class. The problem occurs when there exist methods with the same signature in both the superclasses and subclass.

Que: How to Define a Class in Python? What Is Self? Give An Example Of A Python Class

Ans:

Python is an object-oriented programming language. Unlike procedure-oriented programming, where the main emphasis is on functions, object-oriented programming stresses on objects.

An object is simply a collection of data (variables) and methods (functions) that act on those data. Similarly, a class is a blueprint for that object.

We can think of a class as a sketcueh (prototype) of a house. It contains all the details about the floors, doors, windows, etc. Based on these descriptions we build the house. House is the object.

As many houses can be made from a house's blueprint, we can create many objects from a class. An object is also called an instance of a class and the process of creating this object is called instantiation.

```
class MyNewClass:
    '''This is a docstring. I have created a new class'''
    pass
```

Que: Write a Python class named Rectangle constructed by a length and width and a method which will compute the area of a rectangle

Ans:

```
class Rectangle():
    def __init__(self, l, w):
        self.length = l
        self.width  = w

    def rectangle_area(self):
        return self.length*self.width

newRectangle = Rectangle(12, 10)
print(newRectangle.rectangle_area())
```

Que: Write a Python class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle

Ans:

```
class Circle():

    def __init__(self, r):

        self.radius = r



    def area(self):

        return self.radius**2*3.14



    def perimeter(self):

        return 2*self.radius*3.14



NewCircle = Circle(8)

print(NewCircle.area())
print(NewCircle.perimeter())
```

que:Explain Inheritance in Python with an example? What is init? Or What Is A Constructor In Python?

Ans:

Inheritance is an important aspect of the object-oriented paradigm. Inheritance provides code reusability to the program because we can use an existing class to create a new class instead of creating it from scratch.

In inheritance, the child class acquires the properties and can access all the data members and functions defined in the parent class. A child class can also provide its specific implementation to the functions of the parent class. In this section of the tutorial, we will discuss inheritance in detail.

Que: What is Instantiation in terms of OOP terminology?

Ans:

Instantiate (a verb) and instantiation (the noun) in computer science refer to the creation of an object (or an "instance" of a given class) in an object-oriented programming (OOP) language. Referencing a class declaration, an instantiated object is named and created, in memory or on disk.

Que: What is used to check whether an object o is an instance of class A?

Ans:

Using isinstance() function, we can test whether an object/variable is an instance of the specified type or class such as int or list. In the case of inheritance, we can checks if the specified class is the parent class of an object. For example, isinstance(x, int) to check if x is an instance of a class int.

Que: What relationship is appropriate for Course and Faculty?

Ans:

Association

Que: What relationship is appropriate for Student and Person?

Ans:

inheritance