Assignment-4

Que:1. What is Exception handling in python?

⇒ An exception is an error which happens at the time of execution of a program.

- However, while running a program, python generates an exception that should be handled to avoid your program to crash.
- In python language, exceptions trigger automatically on errors, or they can be triggered and intercepted by your code.
- The exception indicates that, although the event can occur, this type of event happens infrequently.
- When the method is not able to handle the exception, it is thrown to its caller function.
- Eventually, when an exception is thrown out of the main function, the program is terminated abruptly.

* Examples:

- Division by zero.
- Accessing a file which does not exist.
- Addition of two incompatible types.
- Removing the table from the disconnected database server.
- ATM withdrawal of more than the available amount.

Que:2. Explain Try, except and Finally keyword with example.

⇒ An Exception is an Event, which occurs during the execution of the program.
- It is also known as a run time error.

★ Try and except keyword:

- First try clause is executed. i.e. the code between try and except clause.
- If there is no exception, then only try clause will run, except clause will not get executed.
- If any exception occurs, the try clause will be skipped and except clause will run.

Example:

```
def divide (x,y):
    try:
        result = x/y
        print("Yeah ! Your answer is :", result)
    except ZeroDivisionError:
        print("Sorry ! You are dividing by zero")
divide (3,2)
divide (3,0)
```

Output: Yeah ! Your answer is : 1
Sorry ! You are dividing by zero

Finally keyword :

Python provides a keyword finally, which is always executed after try and except blocks.
The finally block always executes after normal termination of try block or after try block terminates due to some exception.

Example :

```
def divide (x, y):
    try:
        result = x // y
    except ZeroDivisionError:
        print("sorry ! you are dividing by zero")
    else:
        print("Yeah ! your answer is :", result)
    finally:
        print('This is always executed')
divide (3,2)
divide (3,0)
```

output :

```
Yeah ! your answer is : 1
This is always executed
Sorry ! you are dividing by zero
This is always executed.
```

Que: 3. Explain different types of file access modes.

⇒ Access mode determines the mode in which the file has to be opened ie. read, write append etc.

r - 1. opens a file for reading only.
2. the file pointer is placed at the begining of the file.
3. This is the default mode.

rb - 1. opens a file for reading only in binary format.
2. this is the default mode.

r+ - 1. opens a file for both reading and writing.

rb+ - 1. opens a file for both reading and writing in binary format.

w - 1. opens a file for writing only
2. overwrites the file if the file exists.

w+ - 1. opens a file for both writing and reading

a - 1. opens a file for appending

ab - 1. opens a file for appending in binary format.

a+ - 1. opens a file for both appending and reading

Que:4. How to open and close .txt file in python?

⇒ There might arise a situation where one needs to interact with external files with Python.

- Python provides inbuilt functions for creating, writing and reading files.
- In this article, we will be discussing how to open an external file and close the same using Python.

★ Opening a file in Python:

There are two types of files that can be handled in python, normal text files and binary files.

Example:

```
file = open("sample.txt")
print(file.read())
```

Output:

```
Hello Geek!
This is a sample text file for the example.
```

★ Closing a file in python:

As you notice, we have not closed any of the files that we operated on in the above examples.

Example:

```
file = open ("sample. txt")
Print (file. read())
file. close()
file. write ("Attempt to write on a closed file!")
```

Output: I/o operation on closed file.

Que:5. How to create MySQL database connection using python.

⇒ In this section of the tutorial, we will discuss the steps to connect the python application to the database.

1. Import mysql. connector module
2. create the connection object
3. create the cursor object
4. Execute the query

★ Creating the connection:

TO create a connection between the MYSQL database and the python application, the connect() method of mysql. connector module is used.

- pass the database details like HostName, userName, and the database password in the method call.

Syntax :

Connection -
object = mysql.connector.connect (host = <host-name> user = <username>, passwd = <password>)

Example :

```
import mysql.connector

myconn = mysql.connector.connect(host = "localhost")
    print (myconn)
```

output :

<mysql.connector.connection.MySQLConnection