Python for Class XII _- Part 10

Topics:

File handling: Need for a data file, Types of file: Text files, Binary files and CSV (Comma separated values) files. *(CSV will be discussed separately)*

Text File: Basic operations on a text file: Open (filename – absolute or relative path, mode) / Close a text file, Reading and Manipulation of data from a text file, Appending data into a text file, standard input /output and error streams, relative and absolute paths.

Binary File: Basic operations on a binary file: Open (filename – absolute or relative path, mode) / Close a binary file, Pickle Module – methods load and dump; Read, Write/Create, Search, Append and Update operations in a binary file. (continued)

Binary files in Python

Binary files are any files where the format isn't made up of readable characters. Binary files can range from image files like JPEGs or GIFs, audio files like MP3s or binary document formats like Word or PDF. In Python, files are opened in text mode by default.

To open files in binary mode, when specifying a mode, add 'b' to it.

```
fobj = open('num.dat', 'rb')
```

Working with Binary Files in Python using pickle module

Python pickle module is used for serializing and de-serializing a Python object structure. Any object in Python can be pickled so that it can be saved on disk. What pickle does is that it "serializes" the object first before writing it to file. Pickling is a way to convert a python object (list, dict, etc.) into a character stream. The idea is that this character stream contains all the information necessary to reconstruct the object in another python script.

```
# Python program to illustrate store
# efficiently using pickle module
# Module translates an in-memory Python object
# into a serialized byte stream—a string of
# bytes that can be written to any file-like object.
```

```python
import pickle

def storeData():
    # initializing data to be stored in db
    Omkar = {'key' : 'Omkar', 'name' : 'Omkar Pathak',
    'age' : 21, 'pay' : 40000}
    Jagdish = {'key' : 'Jagdish', 'name' : 'Jagdish Pathak',

    'age' : 50, 'pay' : 50000}

    # database
    db = {}
    db['Omkar'] = Omkar
    db['Jagdish'] = Jagdish

    # It is important to use binary mode
    dbfile = open('examplePickle.dat', 'ab')

    # source, destination
    pickle.dump(db, dbfile)
    dbfile.close()

def loadData():
    # for reading also binary mode is important
    dbfile = open('examplePickle.dat', 'rb')
    db = pickle.load(dbfile)
    for keys in db:
        print(keys, '=>', db[keys])
    dbfile.close()

if __name__ == '__main__':
    storeData()
    loadData()
```

Output:

```
Omkar => {'key': 'Omkar', 'name': 'Omkar Pathak', 'age': 21, 'pay': 40000}

Jagdish => {'key': 'Jagdish', 'name': 'Jagdish Pathak', 'age': 50, 'pay': 50000}
```

Pickling and Unpickling

So***, pickling*** refers to the process of converting the structures (list, dictionary, etc.) to a byte stream before writing to the file. While reading the contents of the file a reverse process called ***unpickling*** is used to convert the byte stream back to the original structure (list, dictionary, etc.).

How do we use pickle module to work with binary files in Python?

First we need to import pickle module. It provides two main methods – dump() and load().

To write an object to a binary file we use pickle.dump() and to read an object from a binary file we use pickle.load() as illustrated in the following examples:

```
# Python program to write
# a dictionary in binary file
import pickle
def fileOperation():
    dict1 = {"C++" : 1985, "Python" : 1990, "Java" : 1996}

    # Open file for writing in a binary mode
    fobj = open('release.dat', 'wb')
    pickle.dump(dict1, fobj)
    print('Dictionary written to file\n')
    fobj.close()

# Driver's code
fileOperation()
```

Output:

```
Dictionary written to file
```

```
# Python program to read dictionary
# stored in a binary file
import pickle
def main():
    # Open a binary file in read mode
    f = open('release.dat', 'rb')
    # storing data in dictionary release_dates
    release_years = pickle.load(f)
    f.close()
    # Print dictionary
    print(release_years)

if __name__ == "__main__":
    main()
```

Output:

```
{'C++': 1985, 'Python': 1990, 'Java': 1996}
```

```python
# Python program to store integers
# in a binary file and then read
# and display them on the screen
import pickle

def main():

    # Open binary file in write mode
    fobj = open('num.dat', 'wb')
    while True:
        n = int(input("Enter an integer: "))
        pickle.dump(n, fobj)
        choice = input("Do you want to store more integers? (y/n): ")
        if choice.upper() == 'N':
            break
    fobj.close()

    # Reopen the binary file in read mode
    fobj = open('num.dat', 'rb')
    try:
        while True:
            # Storing next integer in the variable n
            n = pickle.load(fobj)
            print(n)
    except EOFError:
        pass
    # close the file
    fobj.close()
```

```python

# Driver's code
if __name__ == "__main__":
    main()
```

Output:

```
Enter an integer: 101
Do you want to store more integers? (y/n): y
Enter an integer: 202
Do you want to store more integers? (y/n): y
Enter an integer: 303
Do you want to store more integers? (y/n): n
101
202
303
```