Binary File Operations in Python

1. Binary Files vs Text Files

- Text Files: Store data in human-readable form (ASCII/Unicode).

- Binary Files: Binary files store data in bytes/binary format (0s and 1s), this data is in non-human-readable format. It is used for images, audio, serialized objects, etc.

2. Opening a Binary File

Python uses different file modes to open binary files:

Mode	Description
`rb`	Read-only (binary)
`rb+`	Read and write (binary)
`wb`	Write-only (binary, creates new file or overwrites existing)
`wb+`	Write and read (binary, creates new file or overwrites existing)
`ab`	Append-only (binary, adds data to end)
`ab+`	Append and read (binary, adds data and allows reading)

Example:

```
file = open("example.dat", "wb")          # Opens in write-binary mode
```

3. Closing a Binary File

Always close a file after operations to free resources.

file.close()

 # Using with statement (recommended):

with open("example.dat", "rb") as file:

data = file.read() # Automatically closes file

4. Pickle Module

The pickle module is used for serializing (converting objects to bytes) and deserializing (converting bytes back to objects).

Methods:

(i)pickle.dump(object, file) → Writes an object to a binary file.

(ii)pickle.load(file) → Reads an object from a binary file.

Example:

import pickle

data = {"name": "Alice", "age": 25}

 # Writing to binary file

with open("data.dat", "wb") as file:

pickle.dump(data, file)

 # Reading from binary file

with open("data.dat", "rb") as file:

loaded_data = pickle.load(file)

print(loaded_data) # Output: {'name': 'Alice', 'age': 25}

5. Operations on Binary Files:

(a) Writing/Creating a Binary File:

```
import pickle
students = [{"roll": 1, "name": "Rahul"}, {"roll": 2, "name": "Priya"}]
with open("students.dat", "wb") as file:
pickle.dump(students, file)
```

(b) Reading from a Binary File:

```
import pickle
with open("students.dat", "rb") as file:
students = pickle.load(file)
print(students)
```

(c) Appending to a Binary File:

```
import pickle
new_student = {"roll": 3, "name": "Amit"}
with open("students.dat", "ab") as file:
pickle.dump(new_student, file)
```

(d) Updating a Binary File :

Method: 01 Read entire file and modify during read and finally write again

```
import pickle
updated_roll = 2
new_name = "Priya Sharma"
with open("students.dat", "rb+") as file:
updated_students = []
try:
while True:
student = pickle.load(file)
if student["roll"] == updated_roll:
student["name"] = new_name
updated_students.append(student)
exceptEOFError:
pass
# Rewrite updated data
with open("students.dat", "wb") as file:
pickle.dump(updated_students, file)
```

Method 2: In-Place Update (More Efficient for Large Files)

If the file is too large, we can update records without loading everything into memory:

```
import pickle

updated_roll = 2

new_name = "Priya Sharma"

with open("students.dat", "rb+") as file:

try:

while True:

pos = file.tell()  # Current position before reading

student = pickle.load(file)

if student["roll"] == updated_roll:

student["name"] = new_name

file.seek(pos)  # Move back to the start of the record

pickle.dump(student, file)  # Overwrite only this record

break  # Exit after update

exceptEOFError:

print("Record not found.")
```

(e) Searching in a Binary File :

```
import pickle

search_roll = 2

with open("students.dat", "rb") as file:
```

```
while True:
try:
student = pickle.load(file)
if student["roll"] == search_roll:
print("Found:", student)
break
exceptEOFError:  # Reached end of file
print("Record not found.")
break
```

IMP Practice Questions:

1. Write a program to store a dictionary in a binary file and then read it back.
2. How is `wb` different from `ab`?
3. Write a program to search for a student’s record by roll number in a binary file.
4. Binary Files vs Text Files.