Experiment No. 6
Serialization in python using Pickle
Date of Performance:
Date of Submission:

Experiment No. 6

Title: Serialization in python using Pickle

Aim: To introduce basic concept of Pickle module

Objective: To introduce data structures in python

Theory:

● What is Serialization?

● Serialization is the process of converting a Python object into a byte stream that can be stored in a file or transmitted over a network.

● What is Pickle?

● Pickle is a Python module used for serializing and deserializing Python objects.

● Why Pickle?

● Pickle provides a convenient way to save Python objects to disk and load them back into memory later.

● How to use Pickle?

● The pickle module provides two main functions: dump() for serialization and load() for deserialization.

1) pickle.dump(obj, file):

The pickle.dump() function is used to serialize a Python object obj and write it to a file specified by the file object file.

This function takes two parameters:

obj: The Python object to be serialized.

file: A file object opened in binary write mode ('wb') where the serialized data will be written.

2) pickle.load(file):

The pickle.load() function is used to deserialize data from a file specified by the file object file and reconstruct the original Python object.

This function takes one parameter:

file: A file object opened in binary read mode ('rb') from which the serialized data will be read and deserialized.

Code: (Serialization)

```
class Emp:
    def __init__(self, Id, name, sal):
        self.Id = Id
        self.name = name
        self.sal = sal
    def display(self):
        print("{:5d}{:20s {:10.2f}}".format(self.Id, self.name, self.sal))



import pickle
import emp
f= open("emp.dat", "wb")
n = int(input("how many emloyees:-"))
for i in range(n):
    Id = int(input("Enter id:-"))
    name = (input("enter name:-"))
    sal = int(input("Enter sal:-"))
    e = emp. Emp(id,name,sal)
    pickle.dump(e,f)
f.close()

with open("emp.dat", "rb") as f:
```

```
    emp_objects = []
    while True:
        emp_obj = pickle.load(f)
        emp_objects.append(emp_obj)
        break


for emp_obj in emp_objects:
    print("ID:", emp_obj.id)
    print("Name:", emp_obj.name)
    print("Salary:", emp_obj.sal)
    print()
```

Output:

how many employees:-1

Enter id:-13

enter name: -Abbdus

Enter sal:-10000

ID: 13

Name: Abbdus

Salary: 10000

Conclusion:

In conclusion, the experiment on serialization in Python using Pickle was successful in demonstrating the ability to easily store and retrieve complex data structures in a serialized format. Pickle proved to be a reliable and efficient tool for this purpose, allowing for seamless encoding and decoding of data without much additional effort.