



| |
|--------------------------------------|
| Experiment No.6 |
| Serialization in python using Pickel |
| Date of Performance: |
| Date of Submission: |



Experiment No. 1

Aim: Serialization in python using Pickle

Objective: To introduce basic concept of Pickle module

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:-

1.

```
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))
```

2.

```
import pickle
import emp
f = open("emp.dat","wb")
n = int(input("How many Employees? :- "))
```

```
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? :- 2
Enter id: 876
Enter name: Shreel Thakur
Enter sal: 5000000
Enter id: 342
Enter name: Bhoomika Surve
Enter sal: 4000000
ID: 876
Name: Shreel Thakur
Salary: 5000000
```

Conclusion:

We would now have a solid understanding of what serialization is, how to use Pickle to serialize Python data structures, and how to optimize Pickle's performance using different arguments and modules.