PICKLING(OBJ SERIALIZATION)

The process of savering or transfering entire object content of main memory into the file of secondary memory by performing single write operation is called pickling.

Pickling concept participates in write operatons.

steps for implementing pickling concept:

a) import pickle module,pickle is a pre-defined module

b) choose file name and open it into a write mode

c) create an iterable obj

d) use dump() of pickle module.dump() save the content form file of main memory into the obj of secondary memory.

syntax: pickle.dump(obj,filepointer)

NOTE: the pickle concept always takes the file in binary format.

UN-PICKLING(OBJ DE-SERIALIZATION)

The process of reading or transfering the entire record content from file of secondary memory into the object of main memory by performing single read operation is called un-pickling.

un-pickilng concept participates in read operations

steps for implementing un-picklling concept:

a) import pickle module

b) choose the file name and open it read mode.

c) use the load() of pickle module.load() is used for transfering or loading the entire record content from file of secondary memory into obj of main memory.

syntax: objname=pickle.load(filepointer)

NOTE: un-pickling concept always takes the file in binary format.

In [ ]:

```
#pickling.py

#program for reading emp detail for kbh and save them into file
import pickle
def saveempdata():
    with open('emp.pick','ab') as fp:
        while(True):
            try:
                print('='*50)
                eno=int(input('enter a emp number'))
                ename=input('enter a emp name')
                sal=float(input('enter a emp sal'))
                dsg=input('enter a emp desigation')
                #create an empty list
                l=list()
                l.append(eno)
                l.append(ename)
                l.append(sal)
                l.append(dsg)
                #write list obj data into file
                pickle.dump(l,fp)
                print('values added sucessfully')
                ch=input('do you want to add more values(yes/no):')
                if(ch=='no'):
                   break
            except ValueError:
                print('do not enterr str,symbols and char')
#main
saveempdata()
```

In [ ]:

```
#un_picklng.py

import pickle
def reademprecords():
    try:
        with open('emp.pick','rb') as fp:
            print('='*50)
            while(True):
                try:
                    obj=pickle.load(fp)
                    for val in obj:
                        print(val)
                except EOFError:
                    print('-------------------------------')
                    break
    except FileNotFoundError :
        print('file does not exist')
#main
reademprecords()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js