```
## Data Abstraction
```

In [1]:

```
# add "__" as prefix to Attributes
```

In [2]:

```
# Example for Data Abstraction
class Employee:
    __count = 0;
    def __init__(self):
        Employee.__count = Employee.__count+1
    def display(self):
        print("The No. of Emplyoyees Count = :",Employee.__count)

emp = Employee()
emp2 = Employee()
```

In [3]:

```
print(emp,emp2)
```

```
<__main__.Employee object at 0x000001E2401F5E80> <__main__.Employee object a
t 0x000001E2401F53A0>
```

In [5]:

```
emp.__count
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-5-f292439264e9> in <module>
----> 1 emp.__count

AttributeError: 'Employee' object has no attribute '__count'
```

In [8]:

```python
# Example -2
class Computer:
    def __init__(self):
        self.__maxprice = 900
    def sell(self):
        print("Selling Price: {}".format(self.__maxprice))
    def setMaxPrice(self, price):
        self.__maxprice = price

# create Obj
c = Computer()
c.sell()

# Change Price Value using Obj
c.__maxprice = 1000
c.sell()

# Change maxprice value through method
c.setMaxPrice(1000)
c.sell()
```

```
Selling Price: 900
Selling Price: 900
Selling Price: 1000
```

In [ ]:

```python
# Built-in Class Attributes
'''
__dict__
__doc__
__name__
__module__
__bases__
'''
```

In [11]:

```python
# Exmaples for Built-in Class Attributes
class Student:
    """This is a Students class to maintain Students Infomation"""
    def __init__(self, name, id,age):
        self.name = name;
        self.id = id
        self.age = age
    def display_details(self):
        print("Name: %s, ID: %d, Age: %d"%self.name,self.id,self.age)

# Create Obj
s =Student("ravi",101,22)
```

In [13]:

```
print(s.__doc__)
print(s.__dict__)
print(s.__module__)
#print(s.__name__)
print(s.__bases__)
```

...

Polymorphism

- Polymorphism contains two words "Poly" and "Morphs"
- one task can performed in different ways

In [14]:

```
# Example
class Parrot:
    def fly(self):
        print("Parrot can Fly")
    def swim(self):
        print("Parrot can't swim")

class Penguin:
    def fly(self):
        print("Penguin can't fly")
    def swim(self):
        print("Penguin can swim")

# CReate Common Interface
def flying_test(bird):
    bird.fly()

# Create Objects
blu = Parrot()
pen = Penguin()

flying_test(blu)
flying_test(pen)
```

```
Parrot can Fly
Penguin can't fly
```

In [15]:

```
# Random Module
import random
```

In [20]:

```
# Generate random integer
print(random.randint(1,10))
```

```
5
```

In [22]:

```
# Generate Random folat value
print(random.random())
```

```
0.6839523509105524
```

In [25]:

```
# Choose/Select random element
list_ex = ['ramu','somu','kiran','ravi','sitha']
random.choice(list_ex)
```

Out[25]:

```
'kiran'
```

In [28]:

```
# Guessing Game
min_num=50
max_num=100
u_range =  input("U required guessing range")
if u_range.lower()=="yes":
    min_num, max_num = int(input("Enter Inital Num= ")),int(input("Enter Max Num= "))
else:
    print("Guessing Num from 50 to 100")
guessing_num = random.randint(min_num,max_num)
n = int(input("Enter Your Guessing number= "))

if guessing_num == n:
    print("You Win")
else:
    print("You Faild")
```

...

In [29]:

```
# os Module
import os
```

In [30]:

```
# print current working Directory/folder
print(os.getcwd())
```

```
C:\Users\HP\Desktop\Python Batch-6
```

In [31]:

```
# print cwd files/folders list
print(os.listdir())
```

```
['.ipynb_checkpoints', 'apssdc.txt', 'college.txt', 'Day-10 [24-9-20]', 'Day
-13 [28-9-20]', 'Day-14 [29-9-20]', 'Day-15 [30-9-20].ipynb', 'Day-3 [16-9-2
0]', 'Day-4 [17-9-20]', 'Day-9 [23-9-20]', 'laptop.txt', 'movie.c', 'movie.m
p3', 'movie.mp4', 'python.txt']
```

In [32]:

```
print(os.listdir("c:\\"))
```

```
['$Recycle.Bin', '$WinREAgent', 'Documents and Settings', 'DumpStack.log.tm
p', 'hiberfil.sys', 'HPSDM', 'inetpub', 'Intel', 'pagefile.sys', 'PerfLogs',
'Program Files', 'Program Files (x86)', 'ProgramData', 'Recovery', 'SoftPaqD
ownloadDirectory', 'swapfile.sys', 'SWSetup', 'System Volume Information',
'system.sav', 'temp', 'Users', 'Windows']
```

In [34]:

```
# change CWD
os.chdir("C:\\Users\\HP\\Desktop\\")
```

In [35]:

```
print(os.getcwd())
```

```
C:\Users\HP\Desktop
```

In [36]:

```
print(os.listdir())
```

```
['.ipynb_checkpoints', 'Current Bills Payments.txt', 'desktop.ini', 'gen1.p
y', 'GoToMeeting.lnk', 'Internet Download Manager.lnk', 'Opera Browser.lnk',
'practice on classes.ipynb', 'Python Batch-6', 'Python Batch-6 [14-9-20 to 3
-10-20] - Shortcut.lnk', 'Quations.txt', 'Roll NUMBERS NOT MATCH.txt', 'Slac
k.lnk', 'Types-of-Inheritance.jpg', 'Untitled.ipynb', 'µTorrent.lnk']
```

In [37]:

```
# Craete folder/derectory
os.mkdir("students class")
```

In [39]:

```
# rename folders/files
os.rename("students class","python class")
```

In [40]:

```
#Remove folder
os.rmdir("python class")
```

In [41]:

```
# Remove file
os.remove("check.txt")
```

In [42]:

```
'.' in "readme.txt"
```

Out[42]:

```
True
```

In [43]:

```
".txt" in "readme.txt"
```

Out[43]:

```
True
```

In [45]:

```
# Check File/Folder by using Address
# os.path.isfile()
print(os.path.isfile("C:\\Users\\HP\\Desktop\\readme.txt"))
print(os.path.isfile("C:\\Users\\HP\\Desktop\\Quations.txt"))
```

```
False
True
```

In [47]:

```
# Check File/Folder by using Address
# os.path.isfolder()
print(os.path.isdir("C:\\Users\\HP\\Desktop\\readme.txt"))
print(os.path.isdir("C:\\Users\\HP\\Desktop\\Quations.txt"))
```

```
True
False
```

In [48]:

```python
# Print folders and files in All childs paths
# os.walk(address)
for path,folders,files in os.walk(input("Enter Location: ")):
    print(path)
    print(folders)
    print(files)
```

```
Enter Location: C:\Users\HP\Desktop\Python Batch-6
C:\Users\HP\Desktop\Python Batch-6
['.ipynb_checkpoints', 'Day-10 [24-9-20]', 'Day-13 [28-9-20]', 'Day-14 [29-9
-20]', 'Day-3 [16-9-20]', 'Day-4 [17-9-20]', 'Day-9 [23-9-20]']
['apssdc.txt', 'college.txt', 'Day-15 [30-9-20].ipynb', 'laptop.txt', 'movi
e.c', 'movie.mp3', 'movie.mp4', 'python.txt']
C:\Users\HP\Desktop\Python Batch-6\.ipynb_checkpoints
[]
['Day-10 [24-9-20]-checkpoint.ipynb', 'Day-13 [28-9-20]-checkpoint.ipynb',
'Day-14 [29-9-20]-checkpoint.ipynb', 'Day-15 [30-9-20]-checkpoint.ipynb', 'D
ay-3 [16-9-20]-checkpoint.ipynb', 'Day-4 [17-9-20]-checkpoint.ipynb', 'Day-9
[23-9-20]-checkpoint.ipynb']
C:\Users\HP\Desktop\Python Batch-6\Day-10 [24-9-20]
[]
['Day-10 [24-9-20] - Jupyter Notebook.pdf', 'Day-10 [24-9-20].ipynb']
C:\Users\HP\Desktop\Python Batch-6\Day-13 [28-9-20]
[]
['Day-13 [28-9-20] - Jupyter Notebook.pdf', 'Day-13 [28-9-20].ipynb']
C:\Users\HP\Desktop\Python Batch-6\Day-14 [29-9-20]
[]
['Day-14 [29-9-20] - Jupyter Notebook.pdf', 'Day-14 [29-9-20].ipynb']
C:\Users\HP\Desktop\Python Batch-6\Day-3 [16-9-20]
[]
['Day-3 [16-9-20] - Jupyter Notebook.pdf', 'Day-3 [16-9-20].ipynb']
C:\Users\HP\Desktop\Python Batch-6\Day-4 [17-9-20]
[]
['Day-4 [17-9-20] - Jupyter Notebook.pdf', 'Day-4 [17-9-20].ipynb']
C:\Users\HP\Desktop\Python Batch-6\Day-9 [23-9-20]
[]
['Day-9 [23-9-20] - Jupyter Notebook.pdf', 'Day-9 [23-9-20].ipynb']
```

In [ ]:

```python
# Hirarchical Inheritance
# Syntax
'''
class Parent:
    parent attr

class Child1(Parent):
    child1 attr

class Child2(Parent):
    Child2 Attri
'''
```

In [49]:

```python
# Ex
class Parent:
    def func1(Self):
        print("This Function from Parent Class")

class Child1(Parent):
    def func2(self):
        print("This Function from Child1 Class")

class Child2(Parent):
    def func3(self):
        print("This Function from Child2 Class")

# Create Objects
ch1 = Child1()
ch2 = Child2()
```

In [51]:

```python
ch1.func1()
ch1.func2()
ch2.func1()
ch2.func3()
```

```
This Function from Parent Class
This Function from Child1 Class
This Function from Parent Class
This Function from Child2 Class
```

In [52]:

```python
# time Module
import time
```

In [54]:

```python
#print current time
print(time.ctime())
print(type(time.ctime()))
```

```
Wed Sep 30 18:33:17 2020
<class 'str'>
```

In [58]:

```python
# Pause the execution using time
time.sleep(5)  #sleep(seconds)
print("Hello Students")
```

```
Hello Students
```

TASKS

- Save Dynamic string to Random text file
- Find dynamic String from Text files

- print number of text files have Dynamic String
- print text files names having dynamic String

In [ ]:

```
1
```