Nested Classes or Inner classes
Class inside class is called nested class or inner class
Defining a class whose implementations used by only one class but not used by another class.

These nested classes can be,
1. Top level classes
2. Local class

Top level class
A class defined inside class is called top level class or nested inner class.

Syntax:
```
class <outer-class-name>:
        class <inner-class-name>:
                variable
                methods
        variables
        methods
```

inner class can be perform operation using data of outer class
inner class can be used to represent objects within outer class or outside the outer class.

```
class A: # ==> Outer class
   class __B: # ==> Inner class
      def m1(self): # object level method
         print("object level method")
   objb=__B() # class level
def main():
   A.objb.m1()
main()
```

Output:
```
object level method
>>>
```
Example:
```
class Person:
   class Address:
      def __init__(self):
```

```
            self.__street=None
            self.__city=None
        def read_address(self):
            self.__street=input("Street:")
            self.__city=input("City:")
        def print_address(self):
            print(f'Street:{self.__street}')
            print(f'City: {self.__city}')
    def __init__(self):
        self.__name=None
        self.__add1=Person.Address()
        self.__add2=Person.Address()
    def read_person(self):
        self.__name=input("Name:")
        self.__add1.read_address()
        self.__add2.read_address()
    def print_person(self):
        print(f'Name: {self.__name}')
        self.__add1.print_address()
        self.__add2.print_address()

def main():
    p1=Person()
    p1.read_person()
    p1.print_person()

main()
```

Output:

```
Name:naresh
Street:s.r.nager
City:hyd
Street:ameerpet
City:hyd
Name: naresh
Street:s.r.nager
City: hyd
Street:ameerpet
City: hyd
>>>
```

Local class
Writing or defining class inside block is called local class.
This class object is created within block(method) but cannot accessible outside block or method.

Syntax:
```
def <method-name>(arg1,arg2,…): #object level/class level/static method
        class <local-class>:
                variables
                methods
```

```
class A:
   def m1(self):
      class B: # local class
         def m2(self):
            print("m2 of B class")
      print("m1 of A")
      objb=B()
      objb.m2()

def main():
   obja=A()
   obja.m1()

main()
```

output:
```
m1 of A
m2 of B class
>>>
```

Python Database Communication (PDBC)
Every application or project required store or save data permanently (OR)
Every application required to persist data.
The data can be saved in permanently using two systems.

1. File System
2. Database System

Limitations of file system

1. Files are not secured because files are managed by operating system
2. Files cannot hold large amount data
3. Files does not provide any query language

Database is a collection of data
Database can hold large amount data
Data stored inside database is secured
Database provides a query language SQL

Database softwares/Applications

1. Oracle
2. MySQL
3. DB2
4. SQL Server
5. MongoDB
6. Sybase
7. postgreSQL