Classes in Python: User defined data types

```
class class_name :
        data members
        methods
```

Example:

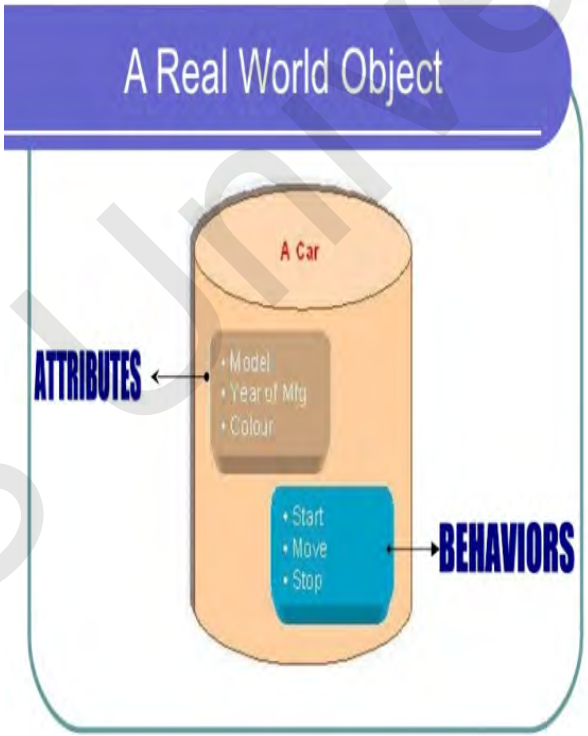
```
class Circle
        radius
        findArea()
        findPerimeter()
```

- Classes provide a means of bundling data and functions together.
- it is a collection of variables and functions
- variables are called as data members and
- Function are called as member functions or **methods**
- Classes are used to represent real world entities

Real world objects/entities have two major things
1)state/attributes(what it is)
2)Behaviour/Actions(what it does)
Pyth0n classes can be used to simulate real world entities

```
class car :
        year; make;
        speed;

        start()
        accelerate();
        brake();
```

```
class flower :
        name;
        color;

        makegarlend()
```



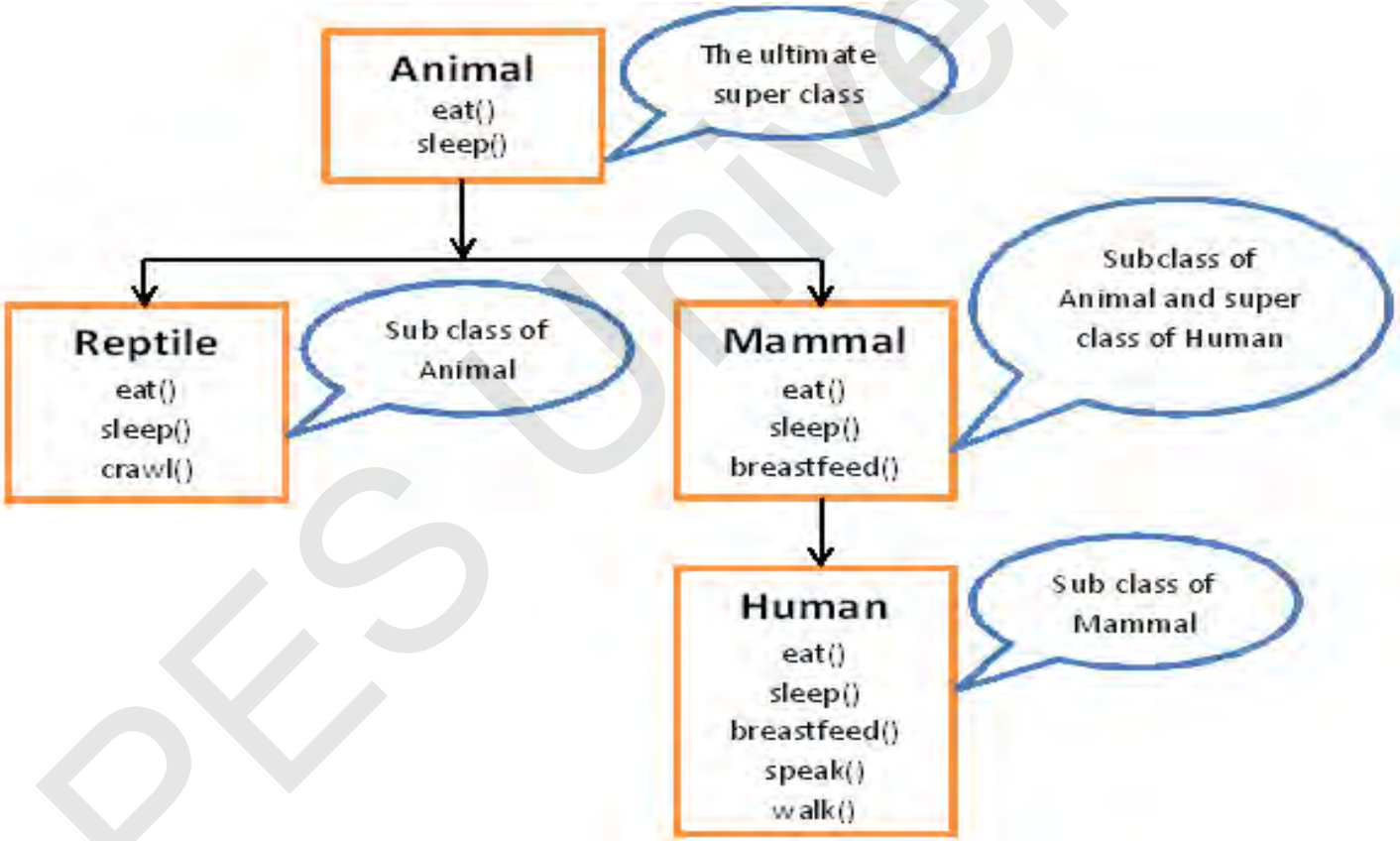
WAP to implement banking operations

```
class Bank:
    def __init__(cust,a,b,c):
        cust.accno=a
        cust.name=b
        cust.bal=c

    def Deposit(self):
        amt=int(input("enter the amount"))
        self.bal=self.bal+amt

    def Wdraw(self):
        amt=int(input("enter the amount"))
        self.bal=self.bal-amt

    def BalEnq(self):
        print("Hello",self.name, "your present bal ",self.bal)

B=Bank(123,"Amar",2000)
B.Deposit()
B.BalEnq()

B.Wdraw()
B.BalEnq()
```

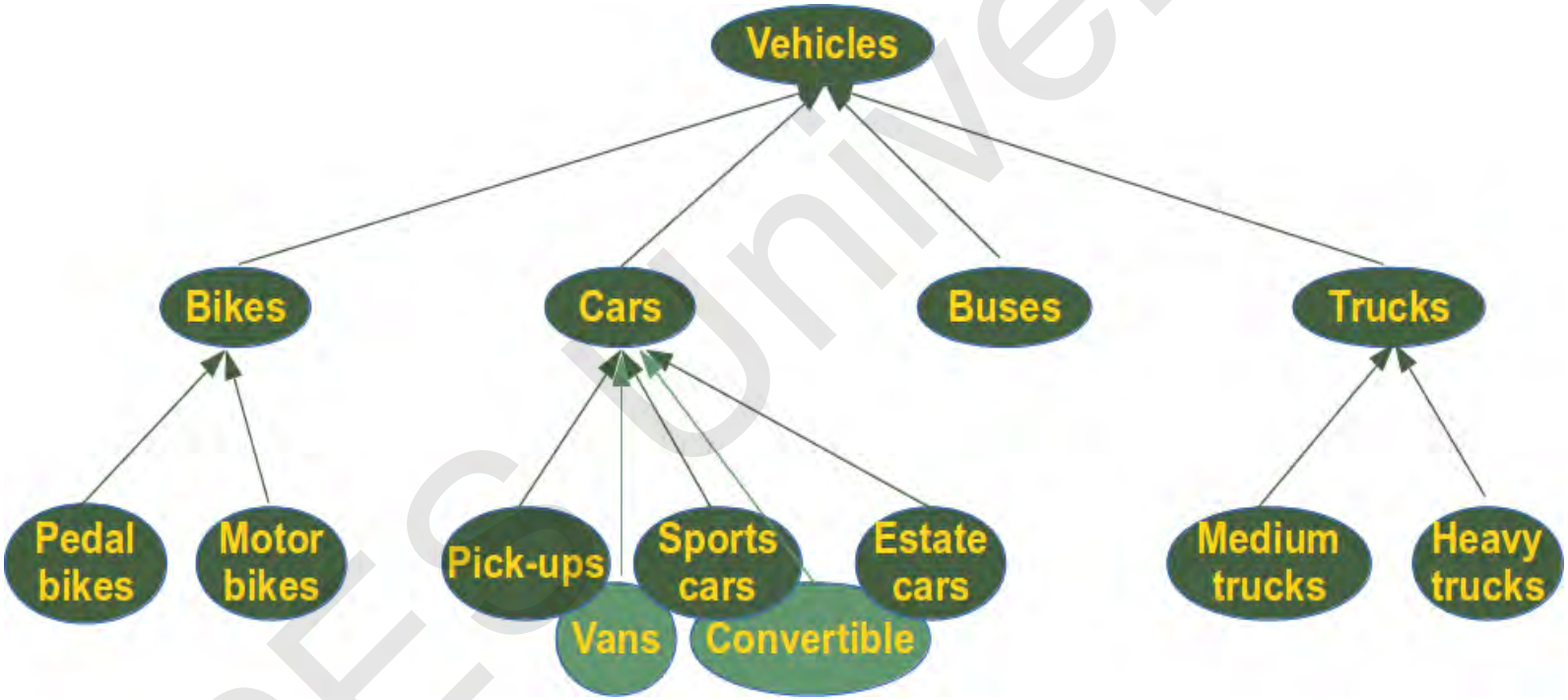
Inheritance

- Inheritance is the process by which one object can acquire the properties of another object.
- Mechanism of deriving a new class from an existing one is called inheritance or derivation.
- The old class is referred to as the base class and the new one is called the derived class.
- It supports the concept of classification

Inheritance in Python cont..



Inheritance in Python cont..



Inheritance in Python cont..

Inheritance

```
class father:
    def __init__(self,a,b,c):
        self.name=a
        self.qual=b
        self.sal=c
    def display(self):
        print(self.name,self.sal)
    def findAnualIncome(self):
        print("income=",12*self.sal)
```

```
class son(father):
    def __init__(self,a,b,c,d):
        father.__init__(self,a,b,c)
        self.sport=d

    def display(self):
        father.display(self)
        print("plays",self.sport)
```

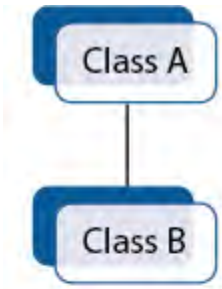
```
f=father("amar","B.E",20000)
f.display()
f.findAnualIncome()
s=son("kumar","B.E",30000,"cricket")
s.display()
s.findAnualIncome()
```

Types of Inheritance:

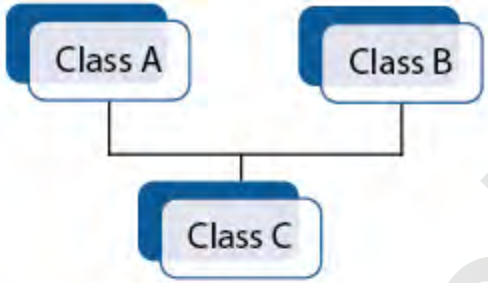
Single Inheritance.

1) Multiple Inheritance.
2) Multilevel Inheritance.
3) Hierarchical Inheritance.
4) Hybrid Inheritance.

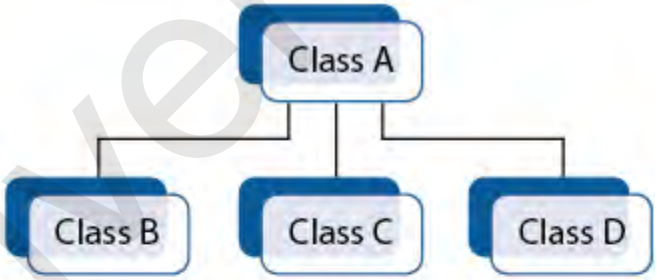
Types of inheritance



Single Inheritance

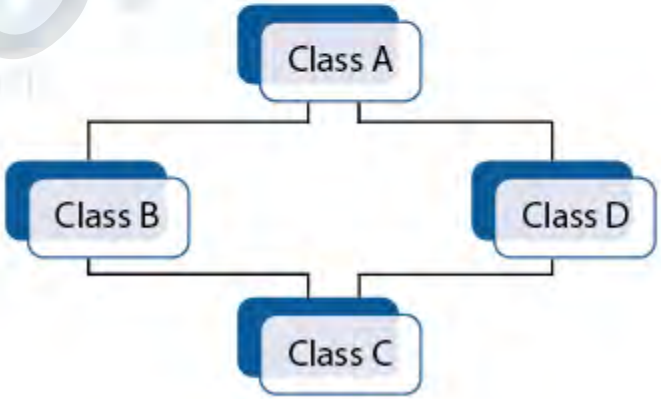


Multiple Inheritance



Hierarchical Inheritance

Multilevel Inheritance



Hybrid Inheritance

Types of inheritance