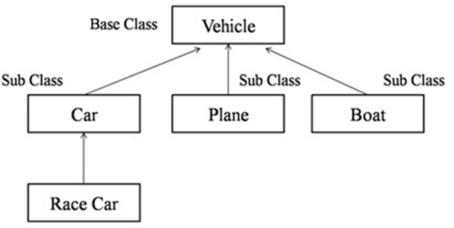
Inheritance

Kelas dapat mewarisi fungsionalitas dari kelas lain. Jika suatu objek dibuat menggunakan kelas yang mewarisi dari superclass, objek tersebut akan berisi metode kelas dan superclass. Warisan adalah ketika kelas menggunakan kode yang dibangun di dalam kelas lain. Jika kita berpikir tentang pewarisan dalam hal biologi, kita dapat memikirkan seorang anak yang mewarisi sifat-sifat tertentu dari orang tua mereka. Artinya, seorang anak dapat mewarisi tinggi atau warna mata orang tua. Anak-anak juga dapat berbagi nama belakang yang sama dengan orang tua mereka. Kelas yang disebut kelas anak atau subclass mewarisi metode dan variabel dari kelas induk atau kelas dasar.



Contoh 1

In [4]:
```
class user:
    def __init__(self,name):
        self.name=name
    def printName(self):
        print("Hello",self.name)
brian=user("Brian")
brian.printName()
```

```
Hello Brian
```

Kita definisikan basic class bernama "user", dan menciptakan objek yg disebut "brian".

In [ ]:
```
class programer(user):
    def __init__(self,name):
        self.name=name
    def dopython(self):
        print("Kelas Programer")
```

Lalu membuat kelas lain yang disebut Programmer. Ini terlihat sangat mirip dengan kelas standar kecuali dari yang diberikan kelas "user" dalam parameter. Ini berarti semua fungsionalitas kelas "user" dapat diakses di kelas "Programmer".

In [3]:
```
class user:
    def __init__(self,name):
        self.name=name
    def printName(self):
        print("Hello",self.name)
class programer(user):
    def __init__(self,name):
        self.name=name
    def doPython(self):
        print("Programming python")
brian=user("Brian")
brian.printName()
Diana=programer("Diana")
Diana.printName()
Diana.doPython()
```

```
Hello Brian
Hello Diana
Programming python
```

Brian adalah turunan dari user dan hanya dapat mengakses metode printName. Diana adalah turunan dari Programmer, kelas dengan warisan dari user, dan dapat mengakses kedua metode dalam Programmer dan user.

Contoh 2

Mari kita buat kelas induk ikan yang nantinya akan kita gunakan untuk membangun jenis ikan sebagai subkelasnya. Masing-masing ikan ini akan memiliki nama depan dan nama belakang di samping karakteristik.

In [ ]:
```
class fish:
    def __init__(self,first_name,last_name="fish",skeleton="bone",eyelids=False):
        self.name=first_name
        self.last=last_name
        self.skeleton=skeleton
        self.eyelids=eyelids
    def swim(self):
        print("The fish is swimming.")
    def swim_backwards(self):
        print("The fish can swim backwards.")
```

Anak atau subclass adalah kelas yang akan diwarisi dari kelas induk. Itu berarti bahwa setiap kelas anak akan dapat menggunakan metode dan variabel dari kelas induk. Sebagai contoh, kelas anak Terryfish yang mensubclasskan kelas Fish akan dapat menggunakan metode swim () yang dideklarasikan dalam Fish tanpa perlu mendeklarasikannya.

In [ ]:
```
class Terryfish(fish):
    pass
```

Kita sekarang dapat membuat objek Trout tanpa harus mendefinisikan metode tambahan

In [2]:
```
class fish:
    def __init__(self,first_name,last_name="fish",skeleton="bone",eyelids=False):
        self.name=first_name
        self.last=last_name
        self.skeleton=skeleton
        self.eyelids=eyelids
    def swim(self):
        print("The fish is swimming.")
    def swim_backwards(self):
        print("The fish can swim backwards.")
class Terryfish(fish):
    pass
terry = Terryfish("Terry")
print(terry.name + " " + terry.last)
print(terry.skeleton)
print(terry.eyelids)
terry.swim()
terry.swim_backwards()
```

```
Terry fish
bone
False
The fish is swimming.
The fish can swim backwards.
```

mari kita buat kelas anak lain yang mencakup metodenya sendiri. Kita akan memanggil kelas ini Clownfish.

In [ ]:
```
class Clownfish(Fish):
    def live_with_anemone(self):
        print("The clownfish is coexisting with sea anemone.")
```

Mari buat objek Clownfish.

In [1]:
```
class fish:
    def __init__(self,first_name,last_name="fish",skeleton="bone",eyelids=False):
        self.name=first_name
        self.last=last_name
        self.skeleton=skeleton
        self.eyelids=eyelids
    def swim(self):
        print("The fish is swimming.")
    def swim_backwards(self):
        print("The fish can swim backwards.")
class Terryfish(fish):
    pass
class Clownfish(fish):
    def live_with_anemone(self):
        print("The clownfish is coexisting with sea anemone.")

casey = Clownfish("Casey")
print(casey.name+ " " + casey.last)
casey.swim()
casey.live_with_anemone()
```

```
Casey fish
The fish is swimming.
The clownfish is coexisting with sea anemone.
```