In [2]:
```
#Single Inheritance

class College(object):
    fees = 1000000
    @classmethod

    def available_cash(cls):
        print(cls.fees)

class MU(College):
        pass

class MU2(College):
        fees = 200000
        @classmethod
        def available_cash(cls):
            print(cls.fees + College.fees)

a = MU()
a.available_cash()

s=MU2()
s.available_cash()
```

```
1000000
1200000
```

In [3]:
```
#Multiple Inheritance

class Dad:
    def hair(self):
        print("Hair is blonde coloured")

class Mom:
    def color(self):
        print("skin colour is cream")

class Child(Dad, Mom):
    pass

c=Child()
print('Child\'s inherited attributes: ')
c.hair()
c.color()
```

```
Child's inherited attributes:
Hair is blonde coloured
skin colour is cream
```

In [6]:
```
class Dinosaur(object):
    def __init__(self):
        self.a = "Dinosaur: Old aged Reptile"
        print(self.a)
        super().__init__()

class Reptile(object):
    def __init__(self):
        self.b = "Reptiles: Hybrids are present day vertebrae, i.e snakes"
        print(self.b)
        super().__init__()

class Velociraptor(Dinosaur,Reptile):
    def __init__(self):
        self.c = "I am an old aged reptile, Velociraptor, a dinosaur"
        print(self.c)
        super().__init__()

o = Velociraptor()
```

```
I am an old aged reptile, Velociraptor, a dinosaur
Dinosaur: Old aged Reptile
Reptiles: Hybrids are present day vertebrae, i.e snakes
```

In [7]:
```
try:
    x = int(input("Enter a number between 100 and 200: "))
    assert x>=5 and x<=10
    print("The entered number: ",x)
except AssertionError:
    print("Condition not satisfied")
finally:
    print("Thank you for working with us")
```

```
Enter a number between 100 and 200: 99
Condition not satisfied
Thank you for working with us
```

In [14]:
```
class AgeValidityException(Exception):
    pass
try:
    age=int(input("Enter your age to check voting eligibility: "))
    if age<18:
        raise AgeValidityException;
    else:
        print("You are eligible to vote!!!")
except AgeValidityException:
        print("You cannot vote....Sorry!!!")
finally:
    print("Thank you...")
```

```
Enter your age to check voting eligibility: 17
You cannot vote....Sorry!!!
Thank you...
```

In [17]:
```
class Error_Exceptions(Exception):
    pass
class ZeroDivision(Error_Exceptions):
    pass
try:
    n = int(input("Enter a number: "))
    if n ==0:
        raise ZeroDivision;
except ZeroDivision:
    print("Input value is zero, try again!")
    print()
```

```
Enter a number: 0
Input value is zero, try again!
```