Multi-level Inheritence

```
class A:
  x = 'class A'
class B(A):
  x = 'class B'
class C(B):
  x = 'class C'
```

```
print(A.x)
```

```
class A
```

```
print(B.x)
```

```
class B
```

```
print(C.x)
```

```
class C
```

```
class A:
  x = 'class A'
class B(A):
  x = 'class B'
class C(B):
  pass
```

```
print(C.x)
```

```
class B
```

```
class A:
  x = 'class A'
class B(A):
  pass
class C(B):
  pass
```

```
print(C.x)
```

```
class A
```

```
print(B.x)
```

```
class A
```

```python
print(A.x)
```

```
class A
```

```python
class A:
  def __init__(self):
    print('class A object created')
class B(A):
  def __init__(self):
    print('class B object created')
class C(B):
  def __init__(self):
    print('class C object created')
```

```python
c1 = C()
```

```
class C object created
```

Multiple-Inheritence

```python
class A:
  x = 'class A'
class B:
  x = 'class B'
class C(A,B):
  x = 'class C'

print(C.x)
```

```
class C
```

```python
class A:
  x = 'class A'
class B:
  x = 'class B'
class C(A,B):
  pass

print(C.x)
```

```
class A
class A:
  pass
class B:
  x = 'class B'
class C(A,B):
  pass

print(C.x)
```

```
class B
```

```
class A:
  def __init__(self):
    print('class A object created')
class B:
  def __init__(self):
    print('class B object created')
class C(A,B):
  def __init__(self):
    print('class C object created')

c1 = C()
```

```
class C object created
```

```
class A:
  def __init__(self):
    print('class A object created')
class B:
  def __init__(self):
    print('class B object created')
class C(A,B):
  pass

c1 = C()
```

```
class A object created
class A:
  pass
class B:
  def __init__(self):
    print('class B object created')
class C(A,B):
  pass

c1 = C()
```

```
class B object created
```

Hirarchical Inheritence

```
class A:
  x = 'class A'
class B(A):
  x = 'class B'
class C(A):
  x = 'class C'
class D(A):
  x = 'class D'
```

Hybrid Inheritence

```
class A:
  x = 'class A'
class B(A):
  x = 'class B'
class C(B):
  x = 'class C'
class D(C,B):
  x = 'class D'

print(D.x)
```

```
class D
```

```
class A:
  x = 'class A'
class B(A):
  x = 'class B'
class C(B):
  x = 'class C'
class D(C,B):
  pass

print(D.x)
```

```
class C
```

```
class A:
  x = 'class A'
class B(A):
  x = 'class B'
class C(B):
  pass
class D(C,B):
  pass

print(D.x)
```

```
class B
```

```
class A:
  x = 'class A'
class B(A):
  pass
class C(B):
  pass
class D(C,B):
  pass

print(D.x)
```

```
class A
```

```
##################################
class A:
  x = 'class A'
class B(A):
  x = 'class B'
class C(A):
  x = 'class C'
class D(B,C):
  x = 'class D'

print(D.x)
```

```
class D
class A:
  x = 'class A'
class B(A):
  x = 'class B'
class C(A):
  x = 'class C'
class D(B,C):
  pass

print(D.x)
```

```
class B
```

```
class A:
  x = 'class A'
class B(A):
  pass
class C(A):
  x = 'class C'
class D(B,C):
  pass

print(D.x)
```

```
class C
```

```
class A:
  x = 'class A'
class B(A):
  pass
class C(A):
  pass
class D(B,C):
  pass

print(D.x)
```

```
class A
```

mro( )

Method resolution order

```
D.mro()
```

```
[__main__.D, __main__.B, __main__.C, __main__.A, object]
```

```
B.mro()
```

```
[__main__.B, __main__.A, object]
```

```
C.mro()
```

```
[__main__.C, __main__.A, object]
```

```
list.mro()
```

```
[list, object]
```

```
A.mro()
```

```
[__main__.A, object]
```

```
dict.mro()
```

```
[dict, object]
```

```
str.mro()
```

```
[str, object]
```

```
int.mro()
```

```
[int, object]
```

```
set.mro()
```

```
[set, object]
```

```
class A:
  x = 'class A'
```

```
dir(A)
```

```
['__class__',
 '__delattr__',
 '__dict__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__le__',
 '__lt__',
 '__module__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__setattr__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 '__weakref__',
 'x']
```

```
dir(object)
```

```
['__class__',
 '__delattr__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__le__',
 '__lt__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__setattr__',
 '__sizeof__',
 '__str__',
 '__subclasshook__']
```

```
len(dir(object))
```

```
23
```

```
len(dir(A))
```

```
27
```

```
print(set(dir(A)) - set(dir(object)))
```

```
{'__module__', 'x', '__weakref__', '__dict__'}
```

```
print(set(dir(list)) - set(dir(object)))
```

```
{'count', '__setitem__', 'index', 'extend', '__len__', '__class_getitem__', 'reverse', '__add_
```

```
print(*list(set(dir(list)) - set(dir(object))),sep='\n')
```

```
    count
    __setitem__
    index
    extend
    __len__
    __class_getitem__
    reverse
    __add__
    __contains__
    __imul__
    clear
    __mul__
    append
    copy
    sort
    __delitem__
    pop
    __iadd__
    remove
    __rmul__
    __reversed__
    __getitem__
    __iter__
    insert
```

✓ 0s completed at 2:50 PM ● ✕