MT21MCS013 Jay Chachapara

Inheritance

Simple / Single Inheritance

In [2]:

```
class superClass :
    def __init__(self) :
        print("Inside super class")

    def setSuperClassData(self) :
        self.__a = input("SuperClassData : ")

    def getSuperClassData(self) :
        print("SuperClassData",self.__a)

class childClass(superClass) :
    def __init__(self):
        superClass.__init__(self)
        print("Inside child class")

    def setChildClassData(self) :
        self.__b = input("ChildClassData : ")

    def getChildClassData(self) :
        print("ChildClassData",self.__b)
if __name__ == "__main__" :
    c1 = childClass()
    c1.setChildClassData()
    c1.setSuperClassData()
    c1.getChildClassData()
    c1.getSuperClassData()
```

```
Inside super class
Inside child class
ChildClassData : 56
SuperClassData : 85
ChildClassData 56
SuperClassData 85
```

Multiple Inheritance

In [3]:

```
class superClass :
    def __init__(self) :
        print("Inside super class")

    def setSuperClassData(self) :
        self.__a = input("SuperClassData : ")

    def getSuperClassData(self) :
        print("SuperClassData",self.__a)

class superClass0 :
    def __init__(self) :
        print("Inside super class 0 ")

    def setSuperClass0Data(self) :
        self.__b = input("SuperClass0Data : ")

    def getSuperClass0Data(self) :
        print("SuperClass0Data",self.__b)

class childClass(superClass,superClass0) :
    def __init__(self):
        superClass.__init__(self)
        superClass0.__init__(self)
        print("Inside child class")

    def setChildClassData(self) :
        self.__c = input("ChildClassData : ")

    def getChildClassData(self) :
        print("ChildClassData",self.__c)

c1 = childClass()

c1.setChildClassData()
c1.setSuperClass0Data()
c1.setSuperClassData()

c1.getChildClassData()
c1.getSuperClass0Data()
c1.getSuperClassData()
```

```
Inside super class
Inside super class 0
Inside child class
ChildClassData : 85
SuperClass0Data : 77
SuperClassData : 99
ChildClassData 85
SuperClass0Data 77
SuperClassData 99
```

Multilevel Inheritance

In [4]:

```
class superClass :
    def __init__(self) :
        print("Inside super class")

    def setSuperClassData(self) :
        self.__a = input("SuperClassData : ")

    def getSuperClassData(self) :
        print("SuperClassData",self.__a)

class IntermediateClass(superClass) :
    def __init__(self) :
        superClass.__init__(self)
        print("Inside Intermediate class")

    def setIntermediateClassData(self) :
        self.__b = input("IntermediateClassData : ")

    def getIntermediateClassData(self) :
        print("IntermediateClassData",self.__b)

class childClass(IntermediateClass) :
    def __init__(self):
        IntermediateClass.__init__(self)
        print("Inside child class")

    def setChildClassData(self) :
        self.__c = input("ChildClassData : ")

    def getChildClassData(self) :
        print("ChildClassData",self.__c)

c1 = childClass()

c1.setChildClassData()
c1.setIntermediateClassData()
c1.setSuperClassData()

c1.getSuperClassData()
c1.getIntermediateClassData()
c1.getChildClassData()
```

```
Inside super class
Inside Intermediate class
Inside child class
ChildClassData : 890
IntermediateClassData : 233
SuperClassData : 212
SuperClassData 212
IntermediateClassData 233
ChildClassData 890
```

Hirarchical Inheritance

In [1]:

```
class superClass :
    def __init__(self) :
        print("Inside super class")

    def setSuperClassData(self) :
        self.__a = input("Enter value of a : ")

    def getSuperClassData(self) :
        print("SuperClassData",self.__a)

class childClass(superClass) :
    def __init__(self):
        superClass.__init__(self)
        print("Inside child class")

    def setChildClassData(self) :
        self.__b = input("Enter value of b : ")

    def getChildClassData(self) :
        print("ChildClassData",self.__b)

class childClass0(superClass) :
    def __init__(self):
        superClass.__init__(self)
        print("Inside child class0")

    def setChildClass0Data(self) :
        self.__c = input("Enter value of c : ")

    def getChildClass0Data(self) :
        print("ChildClass0Data",self.__c)

c1 = childClass()
c2 = childClass0()

print("\nC2")
c2.setChildClass0Data()
c2.setSuperClassData()
c2.getChildClass0Data()
c2.getSuperClassData()

print("\nC1")
c1.setChildClassData()
c1.setSuperClassData()
c1.getChildClassData()
c1.getSuperClassData()
```

```
Inside super class
Inside child class
Inside super class
Inside child class0

C2
Enter value of c : 34
Enter value of a : 12
ChildClass0Data 34
SuperClassData 12

C1
```

```
Enter value of b : 90
Enter value of a : 54
ChildClassData 90
SuperClassData 54
```

Hybrid Inheritance

In [5]:

```
class superClass :
    def __init__(self) :
        print("Inside super class")

    def setSuperClassData(self) :
        self.__a = input("SuperClassData : ")

    def getSuperClassData(self) :
        print("SuperClassData",self.__a)

class IntermediateClass1(superClass) :
    def __init__(self):
        superClass.__init__(self)
        print("Inside Intermediateclass1")

    def setIntermediateClass1Data(self) :
        self.__b = input("IntermediateClass1Data: ")

    def getIntermediateClass1Data(self) :
        print("IntermediateClass1Data",self.__b)

class IntermediateClass2(superClass) :
    def __init__(self):
        superClass.__init__(self)
        print("Inside Intermediateclass2")

    def setIntermediateClass2Data(self) :
        self.__c = input("IntermediateClass2Data : ")

    def getIntermediateClass2Data(self) :
        print("IntermediateClass2Data",self.__c)

class childClass(IntermediateClass1,IntermediateClass2) :
    def __init__(self):
        IntermediateClass1.__init__(self)
        IntermediateClass2.__init__(self)
        print("Inside child class")

    def setChildClassData(self) :
        self.__d = input("ChildClassData : ")

    def getChildClassData(self) :
        print("ChildClassData",self.__d)

c1 = childClass()
c1.setSuperClassData()
c1.setIntermediateClass1Data()
c1.setIntermediateClass2Data()
c1.setChildClassData()

print("\nEntered Data: \n")

c1.getSuperClassData()
c1.getIntermediateClass1Data()
c1.getIntermediateClass2Data()
c1.getChildClassData()
```

```
Inside super class
Inside Intermediateclass1
```

```
Inside super class
Inside Intermediateclass2
Inside child class
SuperClassData : 55
IntermediateClass1Data: 69
IntermediateClass2Data : 584
ChildClassData : 369

Entered Data:

SuperClassData 55
IntermediateClass1Data 69
IntermediateClass2Data 584
ChildClassData 369
```

In [ ]: