Untitled2

May 10, 2023

1 INHERITANCE

1.0.1 It is a process of obtaining data from parent class and transfering to child class(sub_class)

[6]:
```
class parent:

    def test_parent(self):
        print("this is my parent class")
```

[7]:
```
class child(parent):
     pass
```

[8]:
```
child_object = child()
```

[9]:
```
child_object.test_parent()
```

```
this is my parent class
```

[ ]:
```

```

[33]:
```
### Multi lable inheritance : one after another
```

[11]:
```
class class1:

    def test_class1(self):
        print("this is my class1")
```

[12]:
```
class class2(class1):
    def test_class2(self):
        print("this is my class2")
```

[13]:
```
class class3(class2):

    def test_class3(self):
        print("this is my class3")
```

[ ]:
```

```

[14]:
```
obj_class3 = class3()
```

[15]:
```
obj_class3.test_class1()
```

```
this is my class1
```

[16]:
```
obj_class3.test_class2()
```

```
this is my class2
```

[17]:
```
obj_class3.test_class3()
```

```
this is my class3
```

[18]:
```
## class 1 is accesed by 2 and 2 by class 3
```

[ ]:
```

```

1.0.2 Multiple inheritance: it is a process of obtaining the data from two different class to one single class

[ ]:
```
class class5:

    def test_class5(self):
        print("this is my class5")
```

[22]:
```
class class6:
    def test_class6(self):
        print("this is my class6")
```

[23]:
```
class class7(class5,class6):
    pass
```

[27]:
```
obj_class7 = class7()
```

[30]:
```
obj_class7.test_class5()
```

```
this is my class5
```

[31]:
```
obj_class7.test_class6()
```

```
this is my class6
```

[ ]:
```

```

[ ]:
```

```

[ ]: