**LAB 10 (INHERITANCE IN PYTHON)**

**1.Write a Python program that has a class Animal with a method legs(). Create two subclasses Tiger and Dog, access the method leg explicitly with class Dog and implicitly with the class Tiger.¶**

In [27]:

**class** **Animal**:

**def** \_\_init\_\_(self,creature):

self.creature = creature

**def** legs(self):

print("i am a " +self.creature+ " and i have 4 legs")

**class** **Tiger**(Animal):

**pass**

**class** **Dog**(Animal):

**def** legs(self):

Animal.legs(self)

obj1 = Tiger("Tiger")

obj1.legs()

obj2 = Dog("Doberman")

obj2.legs()

i am a Tiger and i have 4 legs

i am a Doberman and i have 4 legs

**2.Write a Python program to create a class Employee. Define two subclasses:Engineer and Manager. Every class should have method named printDesignation() that prints Engineer for Engineer class and Manager for Manager Class.**

In [1]:

**class** **Employee**:

designation = ""

**def** \_\_init\_\_(self,designation):

self.designation = designation

**def** printDesignation(self):

print(self.designation)

**class** **Engineer**(Employee):

**def** \_\_init\_\_(self,designation):

super().\_\_init\_\_(designation)

**def** printDesignation(self):

print(self.designation)

**class** **Manager**(Employee):

**def** \_\_init\_\_(self,designation):

super().\_\_init\_\_(designation)

**def** printDesignation(self):

print(self.designation)

obj1 = Engineer("Engineer")

obj1.printDesignation()

obj2 = Manager("Manager")

obj2.printDesignation()

Engineer

Manager

**3. Write a Python program to demonstrate classes and their attributes.**

In [1]:

**class** **human**(object):

**def** \_\_init\_\_(self,name,age):

self.name = name

self.age = age

**def** intro(self):

print("Hello my name is " +self.name+ " and i am " +str(self.age)+ " years old ")

**class** **person**(human):

**pass**

p = person("AYAAM",20)

p.intro()

Hello my name is AYAAM and i am 20 years old

### 4.Write a Python program to demonstrate Inheritance and method overriding

In [3]:

**class** **square**:

**def** \_\_init\_\_(self,l):

self.length = l

**def** print(self):

print("side of square is", self.length)

**def** area(self):

print("Area of square is", self.length \* self.length)

**class** **rectangle**(square):

**def** \_\_init\_\_(self,l,b):

super().\_\_init\_\_(l)

self.breadth = b

**def** print(self):

print("breadth of rectangle", self.breadth)

**def** area(self):

print("Area of rectangle", self.length \* self.breadth)

p = square(2)

p.print()

p.area()

c = rectangle(4,8)

c.print()

c.area()

side of square is 2

Area of square is 4

side of square is 4

Area of square is 16

**5.Write a Python program to demonstrate multiple Inheritance.**

In [2]:

**class** **parent1**(object):

**def** \_\_init\_\_(self):

self.member1 = "A"

**def** f1(self):

print("I AM PARENT 1 ")

**class** **parent2**(object):

**def** \_\_init\_\_(self):

self.member2 = "B"

**def** f2(self):

print("I AM PARENT 2")

**class** **child**(object):

**def** \_\_init\_\_(self):

parent1.\_\_init\_\_(self)

parent2.\_\_init\_\_(self)

**def** f3(self):

print("I AM THE CHILD")

**def** print\_members(self):

print(self.member1,self.member2)

p1 = parent1()

p1.f1()

p2 = parent2()

p2.f2()

obj = child()

obj.f3()

obj.print\_members()

I AM PARENT 1

I AM PARENT 2

I AM THE CHILD

A B