

④ Create a class circle with a method to compute the area. Initialize the class with the radius

class circle:

def \_\_init\_\_(self, radius):

self.radius = radius

def area(self):

return 3.14 \* self.radius \*\* 2

a = circle(2)

print(a.area())

with methods to compute the area and Perimeter, initialize

① Create a base class Bird with a method fly. Create derived classes Eagle and Penguin. Override the fly method in Penguin to indicate that penguins cannot fly.

class Bird:

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

self.name = name

def fly(self):

print(f"The {self.name} is flying.")

class Eagle(Bird):

pass

class Penguin(Bird):

def fly(self):

print(f"Penguin can't fly")

my\_bird = Bird ("Chetinic Bird")

my\_eagle = Eagle ("Bald Eagle")

my\_penguin = Penguin ("Emperor Penguin")

my\_bird.fly()

my\_eagle.fly()

Q8 Create a class Restaurant with attributes name and menu (a list of items) object. Provide methods to add and remove items from the menu.

Class Item:

def \_\_init\_\_(self, name, price):

self.name = name

self.price = price

def repr(self):

return f'Item (name={self.name}, price={self.price})'

Class Restaurant

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

self.name = name

self.menu = []

def add\_item(self, item):

self.menu.append(item)

```
def remove_item(self, item_name):  
    for item in self.menu:  
        if item.name == item_name:  
            self.menu.remove(item)
```

```
def display_menu(self):  
    if not self.menu:  
        return("The menu is currently empty")
```