**Lab 21**

**Name :-** Aryan Dilipbhai Langhanoja

**Date :-** 23-10-2023

**Enrollment No :-** 92200133030

**CO1: To write, test, and debug simple Python programs**

**CO2: To implement Python programs with conditional, loops and functions**

**Task 1:- Creating class,object and calling method of class**

**Python Code:**

class Phone:

def make\_call(self):

print("Make Phone Call")

def play\_game(self):

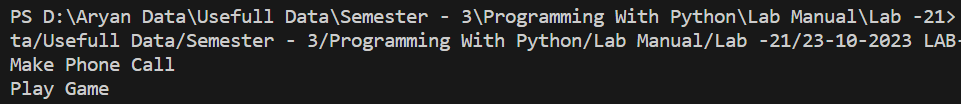
print("Play Game")

phone = Phone()

phone.make\_call()

phone.play\_game()

**Output:**

****

**Task 2:- Providing the values to the attributes of a class**

**Python Code:**

class Phone :

def set\_color(self,color) :

self.color = color

def set\_cost(self,cost) :

self.cost = cost

def show\_color(self) :

return self.color

def show\_cost(self) :

return self.cost

def make\_call(self):

print("Make Phone Call")

def play\_game(self):

print("Play Game")

phone\_1 = Phone()

phone\_1.set\_color('Titanium Grey')

phone\_1.set\_cost('150000')

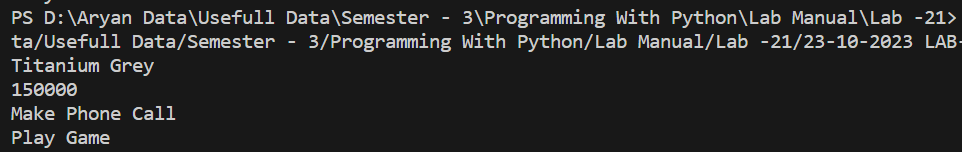
print(phone\_1.show\_color())

print(phone\_1.show\_cost())

phone\_1.make\_call()

phone\_1.play\_game()

**Output:**

****

**Task 3:-** **Creating the constructor of a class**

**Python Code:**

class Employee :

def \_\_init\_\_(self , name , age , salary , gender) :

self.name = name

self.age = age

self.salary = salary

self.gender = gender

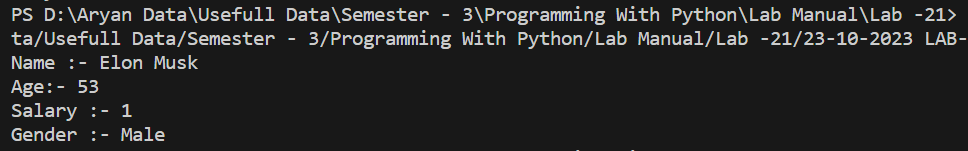
def show\_Employee\_Detail(self) :

print(f"Name :- {self.name}\nAge:- {self.age}\nSalary :- {self.salary}\nGender :- {self.gender}")

employee = Employee("Elon Musk",53,1,"Male")

employee.show\_Employee\_Detail()

**Output:**

****

**Task 4:- Plot of Sinc Function with Numpy and Matplotlib**

**Python Code:**

class Vehicle:

def \_\_init\_\_(self, milage, cost):

self.milage = milage

self.cost = cost

def show\_details(self):

print(f"Vehicle\nMilage :- {self.milage}\nCost :- {self.cost}")

vehicle = Vehicle(65, 85000)

vehicle.show\_details()

class Car(Vehicle):

def \_\_init\_\_(self, milage, cost, types, horse\_power):

super().\_\_init\_\_(milage, cost)

self.types = types

self.horse\_power = horse\_power

def show\_car(self):

super().show\_details()

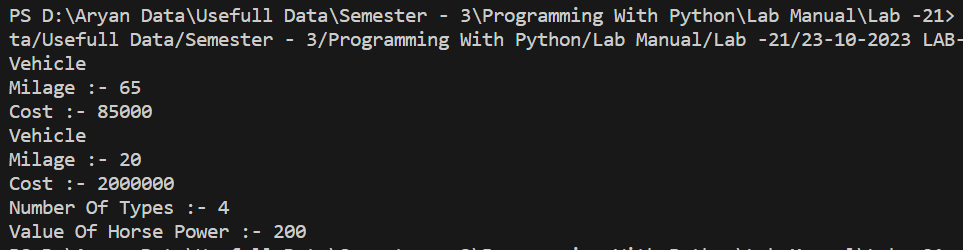
print(

f"Number Of Types :- {self.types}\nValue Of Horse Power :- {self.horse\_power}")

car = Car(20, 2000000, 4, 200)

car.show\_car()

**Output:**



**Post Lab**

**Task 1:- To write a python code for Class and Object**

**Python Code:**

class Person:

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

self.name = name

self.age = age

def introduce(self):

print(f"My name is {self.name}, and I am {self.age} years old.")

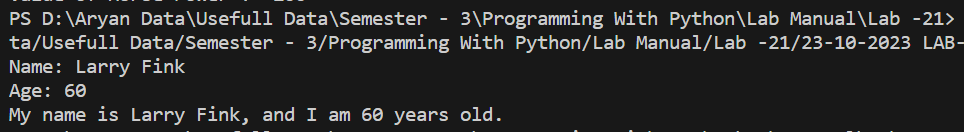
person1 = Person("Larry Fink", 60)

print(f"Name: {person1.name}")

print(f"Age: {person1.age}")

person1.introduce()

**Output:**



**Task 2:- To write a python code for Class and Object**

**Python Code:**

class Animal:

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

self.name = name

def speak(self):

pass

class Dog(Animal):

def speak(self):

return f"{self.name} says Woof!"

class Cat(Animal):

def speak(self):

return f"{self.name} says Meow!"

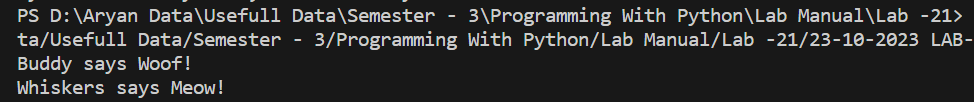
dog = Dog("Buddy")

cat = Cat("Whiskers")

print(dog.speak())

print(cat.speak())

**Output:**

****

**Task 3:- To write a python code for Class and Object**

**Python Code:**

class Employee:

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

self.\_\_name = name

self.\_salary = salary

def get\_name(self):

return self.\_\_name

def set\_name(self, name):

if len(name) > 0:

self.\_\_name = name

def get\_salary(self):

return self.\_salary

def set\_salary(self, salary):

if salary >= 0:

self.\_salary = salary

employee = Employee("John Doe", 50000)

print("Employee Name:", employee.get\_name())

print("Employee Salary:", employee.get\_salary())

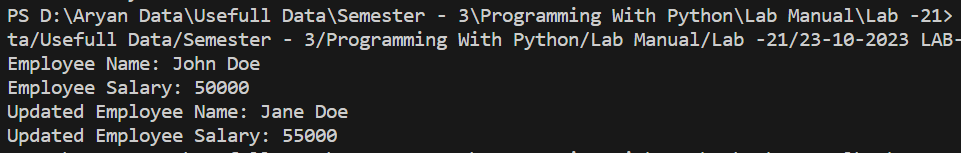
employee.set\_name("Jane Doe")

employee.set\_salary(55000)

print("Updated Employee Name:", employee.get\_name())

print("Updated Employee Salary:", employee.get\_salary())

**Output:**



**Task 4:- To write a python code for Class and Object**

**Python Code:**

class Animal:

def speak(self):

pass

class Dog(Animal):

def speak(self):

return "Woof!"

class Cat(Animal):

def speak(self):

return "Meow!"

class Bird(Animal):

def speak(self):

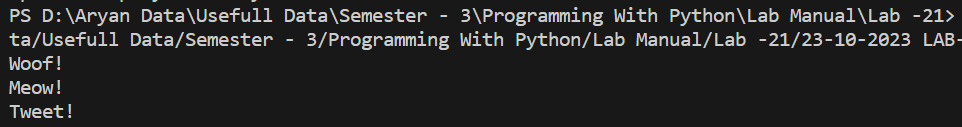
return "Tweet!"

animals = [Dog(), Cat(), Bird()]

for animal in animals:

print(animal.speak())

**Output:**

****