NAME:MANVITH BALAJI

SECTION:A

MIS NO. :112315115

1.

Input code:

class A:

    def action(self):

        print("A")

class B(A):

    def action(self):

        print("B")

class C(A):

    def action(self):

        print("C")

class D(B,C):

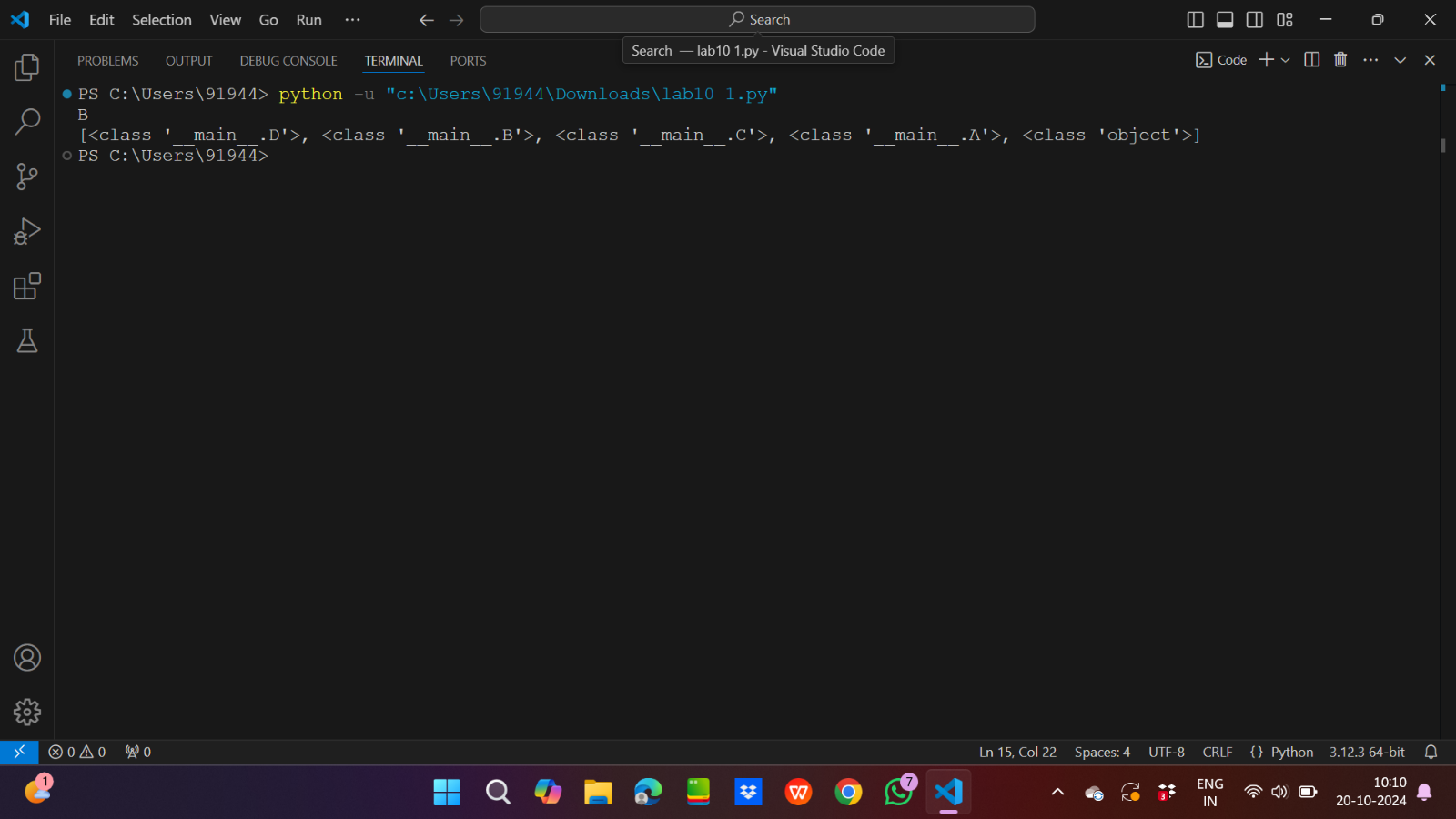
    def \_\_init\_\_(self):

        self.action()

d=D()

print(D.mro())

Output:



2.

Input code:

class A:

    def action(self):

        print("A")

class B(A):

    def action(self):

        super().action()

        print("B")

class C(A):

    def action(self):

        super().action()

        print("C")

class E(A):

    def action(self):

        super().action()

        print("E")

class D(B,C,E):

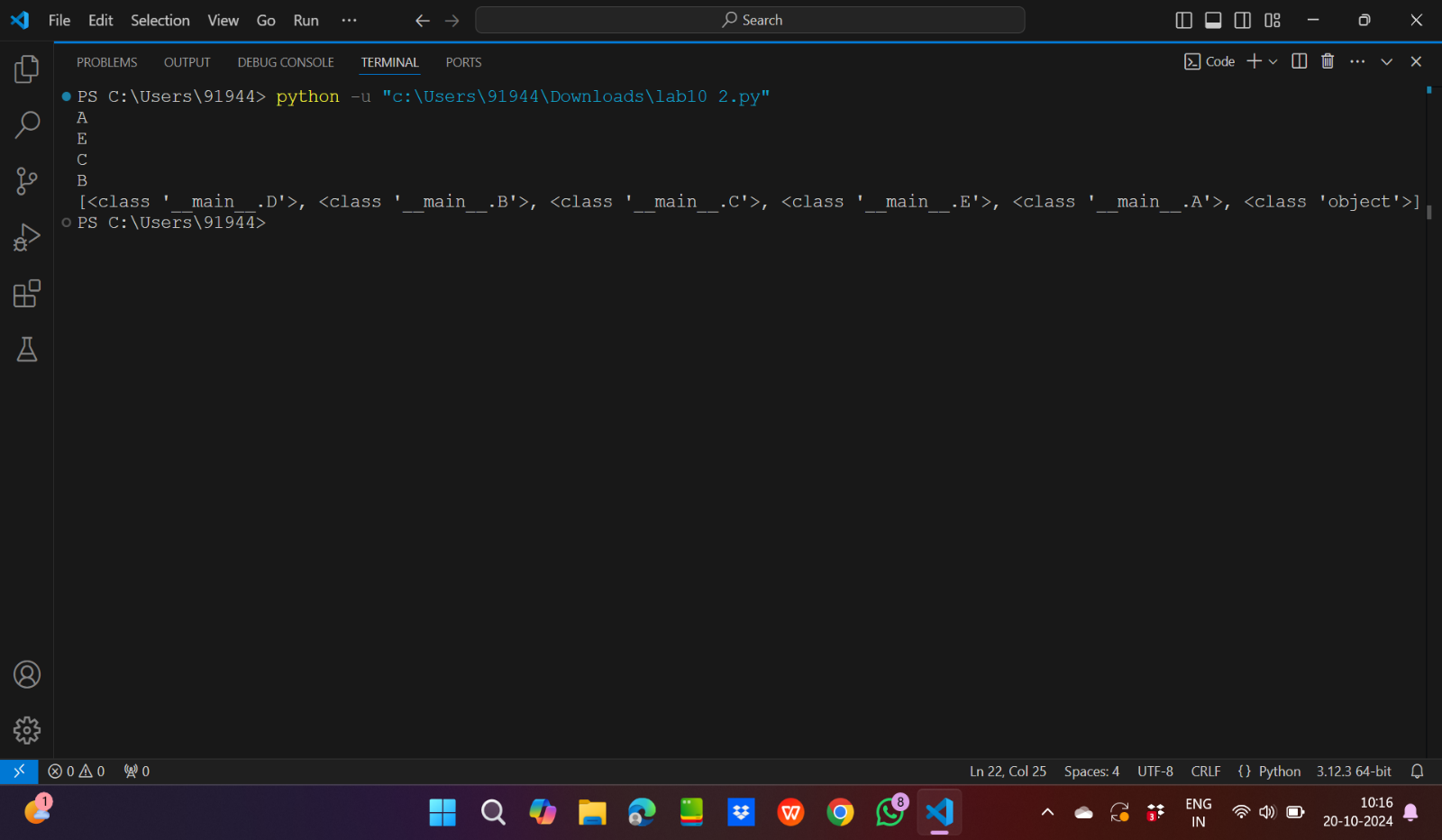
    def action(self):

        super().action()

d=D()

d.action()

print(D.mro())

Output: 

3.

Input code:

class Person:

    def \_\_init\_\_(self,name):

        self.name=name

class Employee(Person):

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

        super().\_\_init\_\_(name,salary,department)

class Manager(Person):

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

        super().\_\_init\_\_(name)

        self.salary=salary

        self.department=department

class Director(Employee,Manager):

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

        super().\_\_init\_\_(name,salary,department)

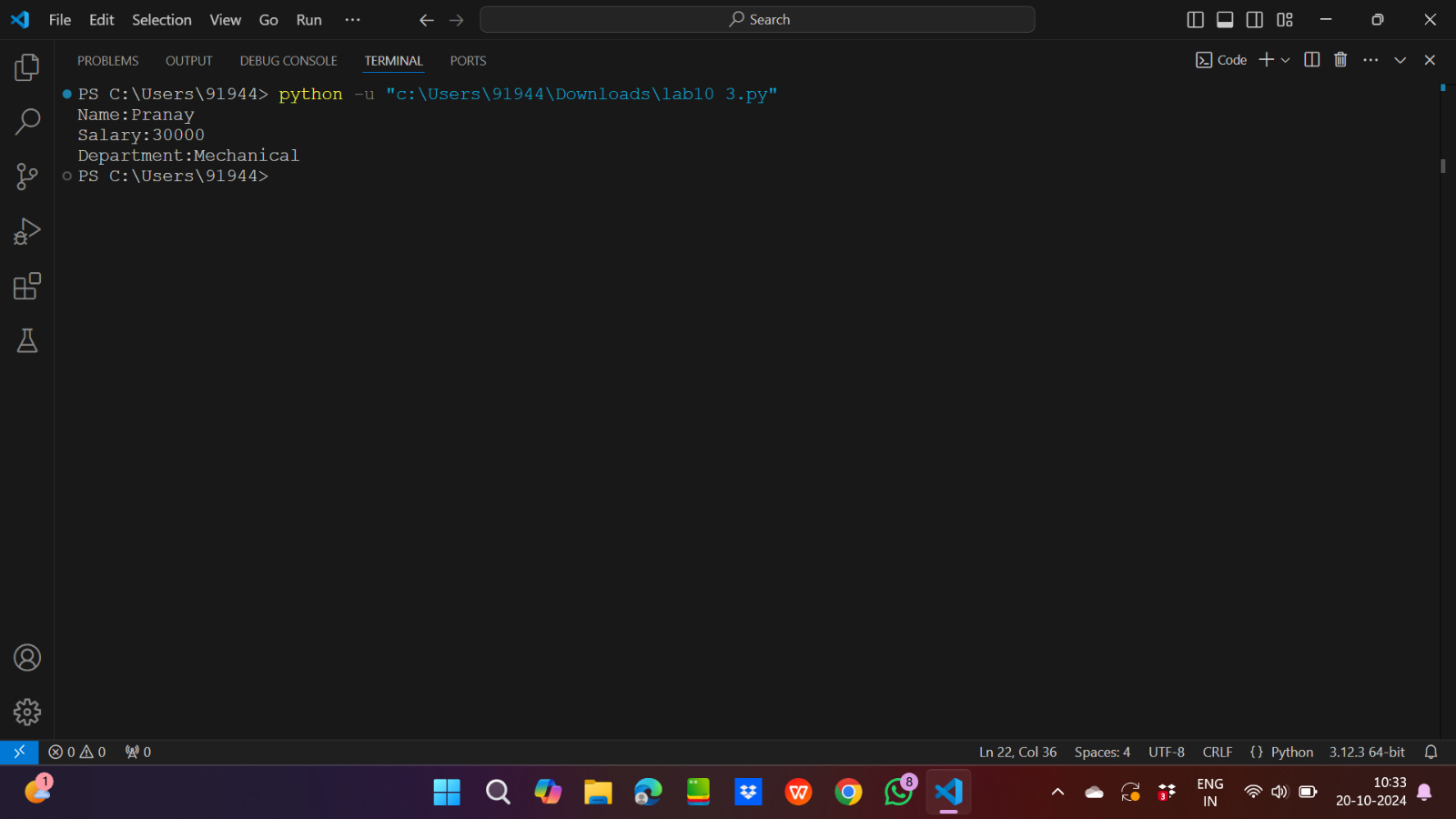
d=Director("Pranay",30000,"Mechanical")

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

print(f"Salary:{d.salary}")

print(f"Department:{d.department}")

Output:



4.

Input code:

from random import \*

class Base1:

    def process(self):

        print("Base1")

class Base2:

    def process(self):

        print("Base2")

class Base3:

    def process(self):

        print("Base3")

mylist=[Base1,Base2,Base3]

a=randint(0,2)

class Derived(mylist[a],mylist[(a+1)%3],mylist[(a+2)%3]):

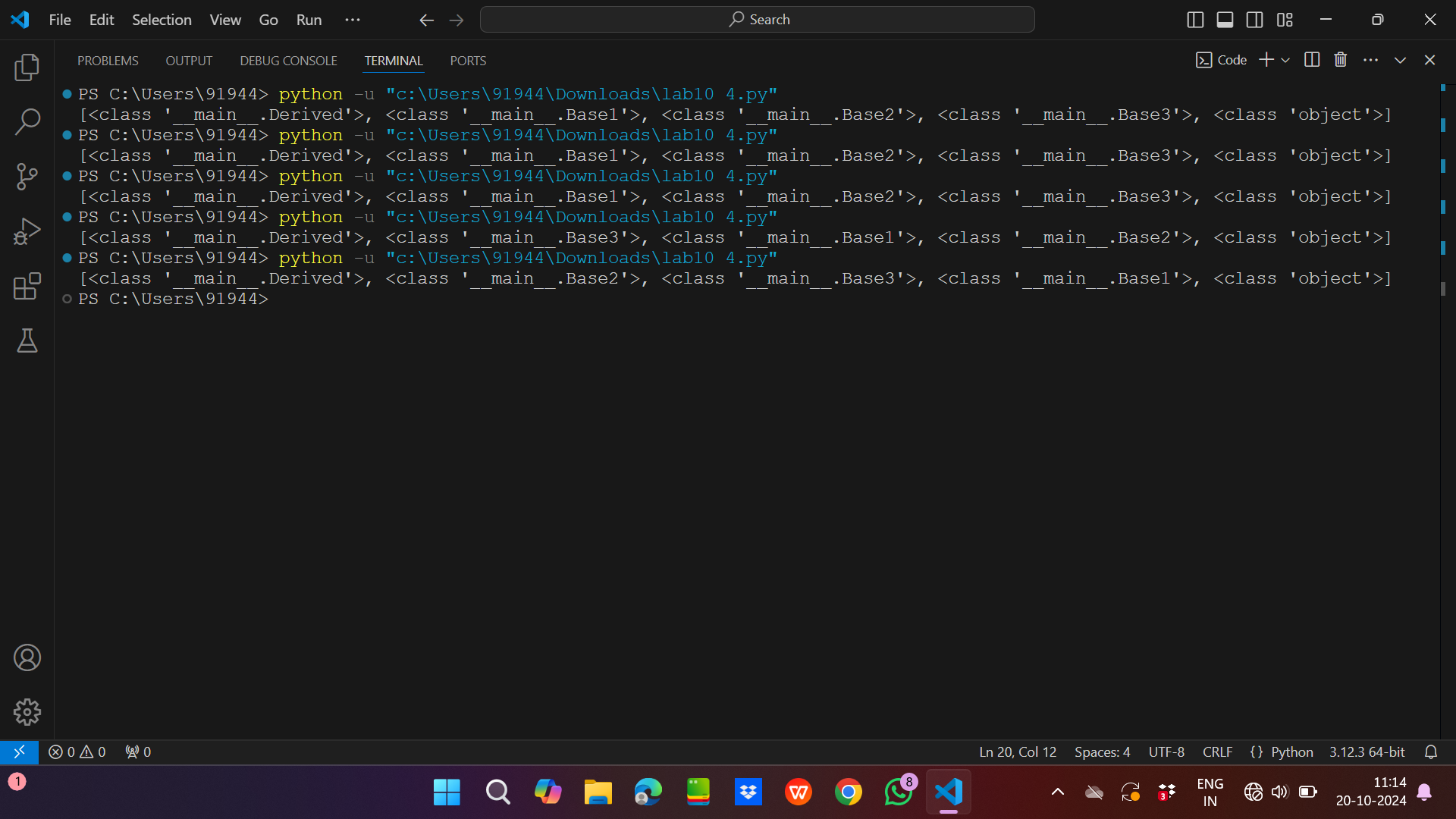
    def process(self):

        print("Derived")

d=Derived()

print(Derived.mro())

Output:



5.

Input code:

class X:

    def greet(self):

        print("Hello from X")

class Y:

    def greet(self):

        print("Hello from Y")

class Z(X,Y):

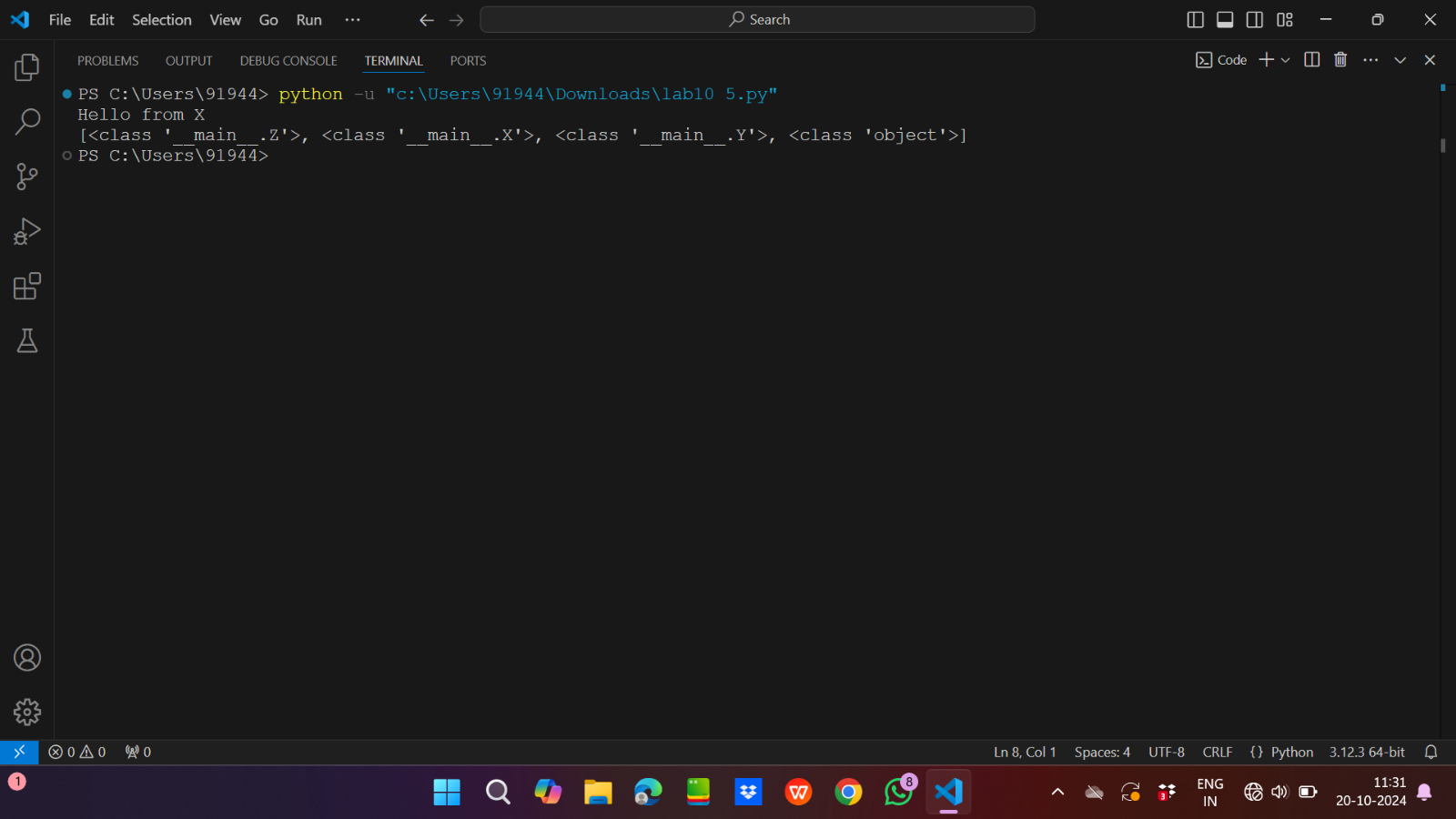
    def \_\_init\_\_(self):

        super().greet()

z=Z()

print(Z.mro())

Output:



6.

Input code:

class Base1:

    data=1

    def info(self):

        super().info()

        print(f"Base1 and data:{Base1.data}")

class Base2:

    data=2

    def info(self):

        super().info()

        print(f"Base2 and data:{Base2.data}")

class Base3:

    data=4

    def info(self):

        print(f"Base3 and data:{Base3.data}")

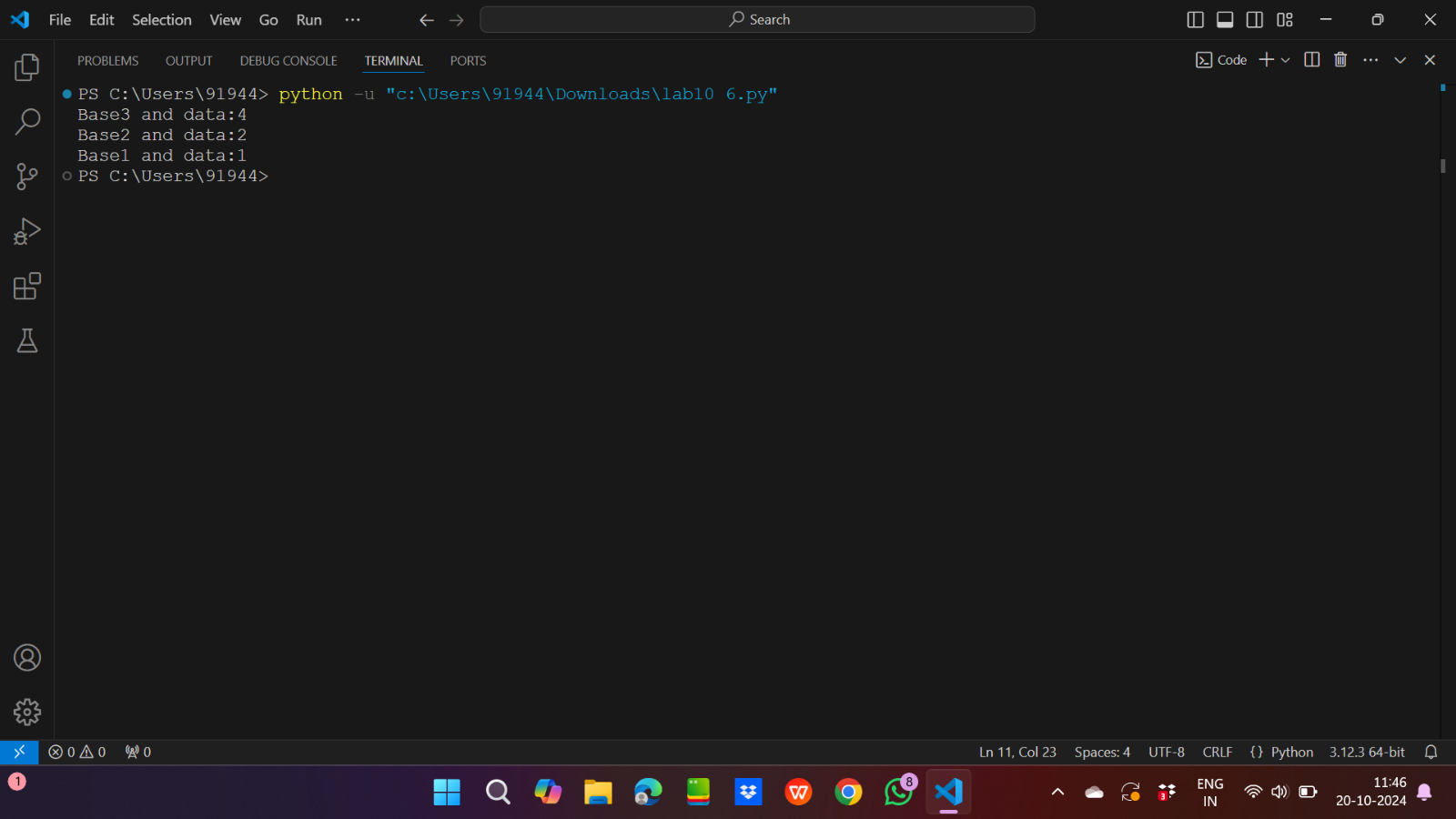
class Combined(Base1,Base2,Base3):

    def \_\_init\_\_(self):

        super().info()

a=Combined()

Output:



7.

Input code:

class Person:

    def \_\_init\_\_(self,name):

        self.name=name

class Employee(Person):

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

        super().\_\_init\_\_(name,department)

        self.salary=salary

class Manager(Person):

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

        super().\_\_init\_\_(name)

        self.department=department

class Director(Employee,Manager):

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

        super().\_\_init\_\_(name,salary,department)

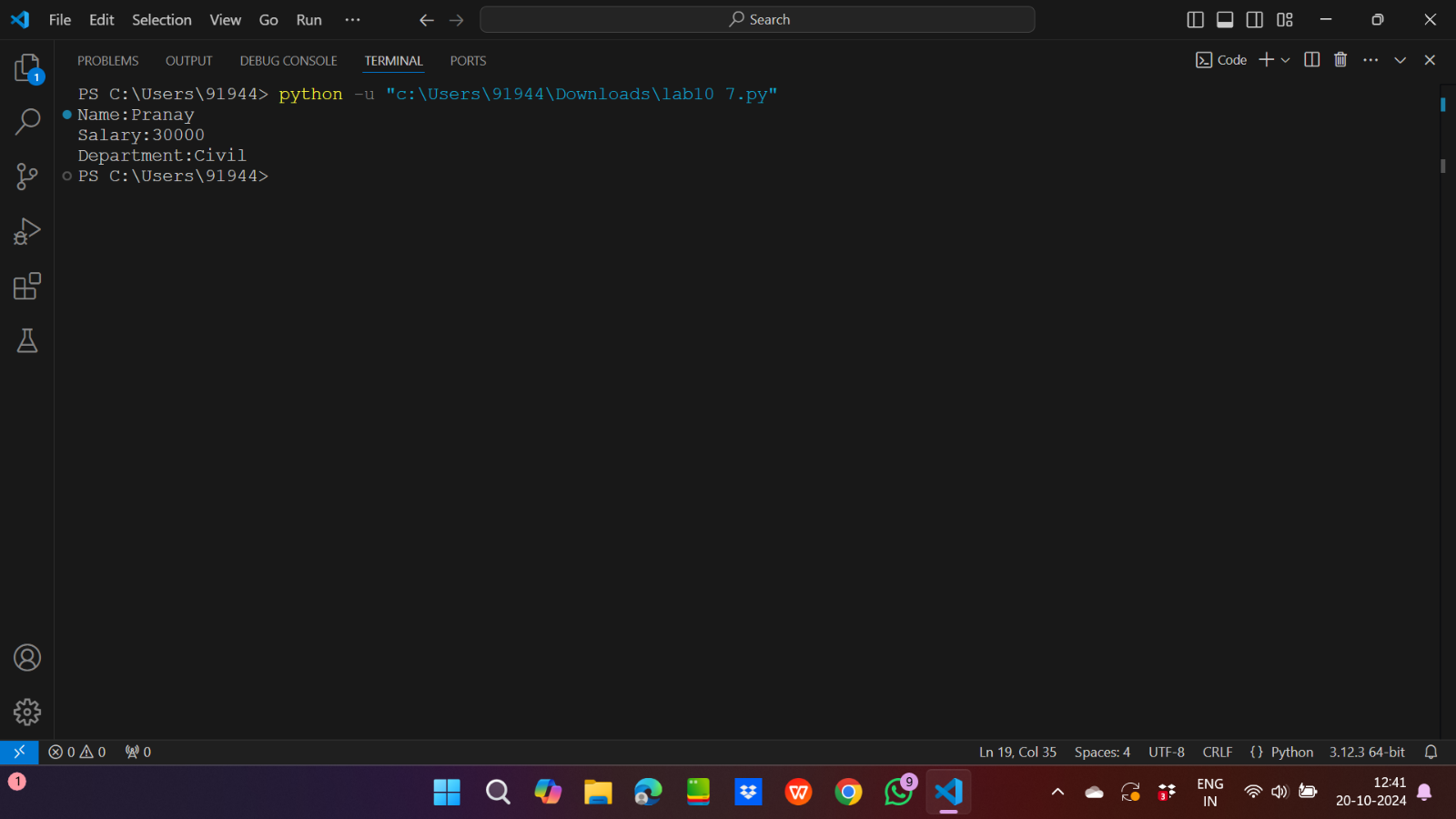
d=Director("Pranay",30000,"Civil")

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

print(f"Salary:{d.salary}")

print(f"Department:{d.department}")

Output:



8.

Input code:

class Employee:

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

        self.name=name

        self.age=age

        self.\_\_salary=salary

    def set\_salary(self,new\_salary):

        if new\_salary>0:

            self.\_\_salary=new\_salary

        else:

            print("Salary cannot be negative")

    def get\_salary(self):

        if sa>0:

            print(f"The current salary is {self.\_\_salary}")

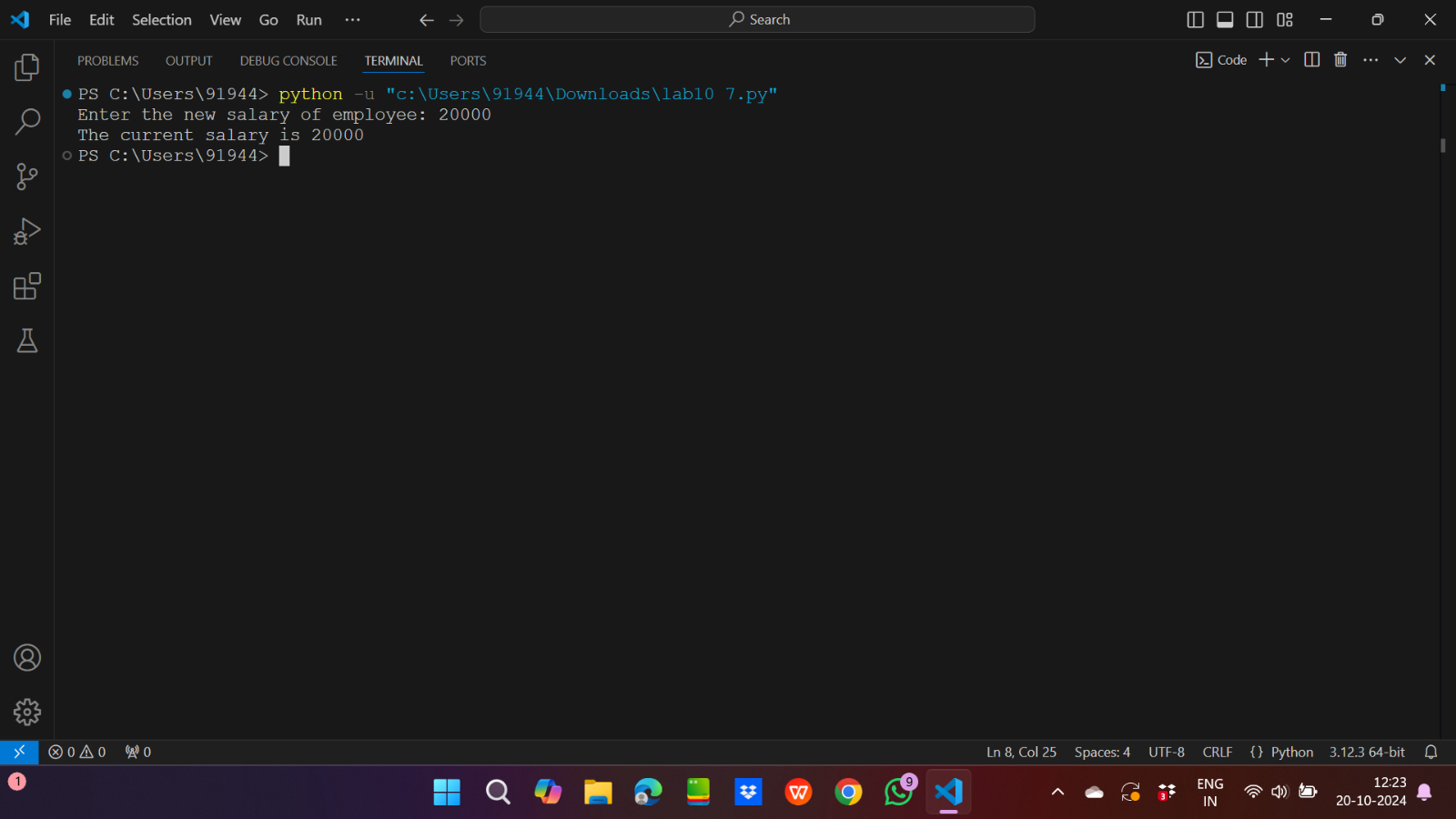
sa=int(input("Enter the new salary of employee: "))

obj=Employee("Pranay",17,30000)

obj.set\_salary(sa)

obj.get\_salary()

Output:



9.

Input code:

class Student:

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

        self.name=name

        self.roll\_no=roll\_no

    def set\_marks(self,marks):

        if ma>0 and ma<100:

            self.\_\_marks=marks

        else:

            print("Marks should be between 0 to 100")

    def get\_marks(self):

        if ma>0 and ma<100:

            print(f"Marks:{self.\_\_marks}")

ma=int(input("Enter the marks of the student: "))

obj=Student("Pranay",14)

obj.set\_marks(ma)

obj.get\_marks()

Output:

