From abc import ABC, abstractmethod

# Abstract Class

Class AbstractMethods(ABC):

Def welcome(self):

Pass

Def test\_data(self, name):

Pass

Def count\_vowels(self, name):

Pass

Def calculate\_grade(self, name, mark1, mark2):

Pass

# Derived Class

Class UserMethods(AbstractMethods):

Def welcome(self):

Print(“Welcome To WTS!\nWe wish you the Best!!”)

Def test\_data(self, name):

Print(f”Welcome {name}!\nHave a nice day!”)

Def count\_vowels(self, name):

Vowels = “aeiouAEIOU”

Count = {v: name.count(v) for v in vowels if v in name}

Total\_vowels = sum(count.values())

Print(f”Count of Vowels are: {total\_vowels}”)

For vowel, cnt in count.items():

Print(f”{vowel}: {cnt}”)

Def calculate\_grade(self, name, mark1, mark2):

Total\_marks = mark1 + mark2

If total\_marks > 95:

Grade = “E”

Elif 80 <= total\_marks < 95:

Grade = “A+”

Elif 75 <= total\_marks < 80:

Grade = “A”

Elif 60 <= total\_marks < 75:

Grade = “B”

Else:

Grade = “F”

Print(f”{name}’s Grade: {grade}”)

# Example Usage

User = UserMethods()

User.welcome()

Name\_input = input(“Please input a name: “)

User.test\_data(name\_input)

User.count\_vowels(name\_input)

Mark1 = int(input(“Enter Mark1: “))

Mark2 = int(input(“Enter Mark2: “))

User.calculate\_grade(name\_input, mark1, mark2)

Def check\_password():

Password = input(“Enter Your Input: “)

If any(char.isdigit() for char in password):

Print(“Your Output will Break here – Welcome”)

Else:

Filtered\_password = ‘’.join([char for char in password if not char.isdigit()])

Print(f”Your output will Continue – {filtered\_password}”)

Check\_password()