Week 1:

Task 3:

A company calculates the monthly electricity bill based on the following rules:

Inputs

1.   units_consumed(int)

2.   is_senior_citizen(bool)

3.   has_solar_panel(bool)

4.   payment_mode(str-'online' or 'offline')

Billing rules:

1.   Base charge per unit

First 100 units – 3rs per unit

Next 200 units(101-300) – 5rs per unit

Above 300 units – 8rs per unit

2.   Senior citizen discount

If is_senior_citizen==true □10% discount on the bill

3.   Solar panel benefit

If has_solar_panel==true and units>250□ 300rs flat discount

If has_solar_panel==true and units<=250□ 500rs flat discount

4.   Payment mode surcharge

If payment_mode=='offline' and total_bill>=1000rs□ 100rs surcharge

If payment_mode=='online'□ no surcharges

5.   Minimum payable amount

Final bill cannot be less than 200rs

Task:

1. Calculate the electricity bill based on the following conditions

2. Apply all discounts and surcharges

3. Ensure minimum payable amount is enforced

4. Print the final bill amount

Use Cases (Test Scenarios):

Use Case 1

units consumed = 180

Is_senior_citizen = True

has_solar_panel = True

payment_mode = "online"

Expected Logic

Tiered hilling (100 + 80 units)

Senior citizen discount

Solar discount ($250 units)

No surcharge

Final bill > ₹200

Use Case 2

units_consumed = 320

is_senior_citizen = False has_solar_panel = True

payment_mode = "offline"

Expected Logic

Highest slab applied Solar discount (>250 units)

Offiine payment surcharge (bill ₹1000)

No senior discount

Use Case 3

units_consumed = 90

is senior citizen - True

has_solar_panel = False

payment mode "offline"

Expected Logic

Low consumption

Senior discount

Offline surcharge (bill < ₹1000)

Enforce minimum bill ₹200

Program (electricity_bill.py):

```python
#get the inputs
units_consumed=int(input('enter units consumed:'))
is_senior_citizen=input('is the user senior citizen(y/n):')
has_solar_panel=input('does the user have solar panel(y/n):')
payment_mode=input('enter mode of payment(online/offline):')

#calculate base charge per unit
bill=0
if units_consumed<=100:
    bill=units_consumed*3
    print('base charge:', bill)
elif(units_consumed>100 and units_consumed<=300):
    bill=(100*3)+((units_consumed-100)*5)
    print('base charge:', bill)
elif(units_consumed>300):
```

```python
    bill=(100*3)+(200*5)+((units_consumed-300)*8)
    print('bill for base charge:', bill)


#check senior citizen discount
if is_senior_citizen=='y' or is_senior_citizen=='Y':
    dis=bill*0.10
    bill=bill-dis
    print('bill after senior citizen discount:', bill)


#check solar panel benefit
if (has_solar_panel=='y' or has_solar_panel=='Y') and (units_consumed<=250):
    bill=bill-500
    print('bill after checking solar panel discount:',bill)
elif (has_solar_panel=='y' or has_solar_panel=='Y') and (units_consumed>250):
    bill=bill-300
    print('bill after checking solar panel discount:',bill)


#calculate payment mode surcharge
if (payment_mode=='offline' or payment_mode=='OFFLINE' or payment_mode=='Offline') and bill<1000:
    bill=bill+250
    print('bill after adding payment mode surcharges:',bill)
elif (payment_mode=='offline' or payment_mode=='OFFLINE' or payment_mode=='Offline') and bill>=1000:
    bill=bill+100
    print('bill after adding payment mode surcharges:',bill)
elif(payment_mode=='online' or payment_mode=='ONLINE' or payment_mode=='Online'):
    print('bill after adding payment mode surcharges:',bill)


#check the valid minimum bill
```

```python
if(bill>200):

    print('total electricity bill:', bill)
else:

    print('total electricity bill:', 200)
```

Output: