Weekly Homework: PM2 + SAC M1

Name: _____ Date: _____

Control Flow (if/elif/else, loops, boolean logic) and SD theory (IPO charts, data dictionaries, functional/non-functional requirements). Solar theme. ***Due: next class.***

Part A: Read the Code

/10

Write what each program outputs. Trace by hand — no computer.

A1. Conditionals

4 marks

```
system_kw = 6.6

if system_kw > 10:
    print("Commercial")
elif system_kw > 5:
    print("Standard")
else:
    print("Small")
```

Output:

```
bill = 350
has_solar = False
owns_home = True

if bill > 300 and owns_home:
    if not has_solar:
        print("Apply now!")
    else:
        print("Already set")
else:
    print("Not eligible")
```

Output:

A2. Loops

4 marks

```
total = 0
for year in range(1, 4):
    total = total + 1200
    print(f"Year {year}: ${total}")
```

Output:

```
cost = 5000
year = 0
while cost > 0:
    cost = cost - 1800
    year = year + 1
print(f"Paid off: {year} yrs")
```

Output:

Hint: trace `cost` after each iteration.

A3. Spot the bugs

2 marks

Each snippet has **one bug**. Circle it and write the fix.

```
tier = input("Tier: ")
if tier = "budget":
    price = 1000
```

Fix:

```
total = 0
for i in range(1, 6)
    total = total + i
```

Fix:

A4. Combined logic

4 marks — bonus

```
bills = [400, 550, 300]
for bill in bills:
    savings = bill * 4 * 0.7
    if savings > 1200:
        print("High")
    else:
        print("Moderate")
```

Output:

```
x = 15
if x > 20:
    print("A")
elif x > 10 and x < 20:
    print("B")
elif x == 15:
    print("C")
else:
    print("D")
```

Output:

Why doesn't `"C"` print?

Part B: Write the Code /12

B1. Solar panel rating 3 marks

Ask for a panel efficiency (0–100) and print: 90+: `"Excellent"`, 75–89: `"Good"`, 60–74: `"Average"`, below 60: `"Poor"`. Use `if/elif/else`.

B2. Input validation 3 marks

Write a `while` loop that asks for the number of solar panels (must be 1–50). Print an error and re-ask if out of range.

B3. Compound growth projection 3 marks

A solar system saves $1,200 in year 1, but electricity prices rise 3% each year (compound growth). Use a `for` loop for years 1–5. Calculate each year's savings using `savings = 1200 * 1.03 ** year` and print the year and savings. Include a `#` comment explaining the ** operator.

B4. Rebate eligibility 3 marks

A household qualifies if: income < $180,000 `and` owns home `and not` already has solar. Store three values in variables, use a **single** `if` with `and`/`not`. Print `"Qualified"` or `"Not qualified"`.

Part C: Software Development Theory /18

C1. Problems, needs, opportunities 3 marks

A school wants to reduce electricity costs. Teachers manually read the meter monthly and record values in a spreadsheet. They cannot compare costs before and after installing solar. The deputy principal spends hours each term creating reports.

Problem	
Need	
Opportunity	

C2. Functional vs Non-functional requirements 4 marks

Write **F** (functional) or **NF** (non-functional) next to each requirement.

#	Requirement	F/ NF
1	The system shall calculate payback period from cost and annual savings.	
2	All dollar values shall be displayed to 2 decimal places.	
3	The system shall accept the user's quarterly electricity bill as input.	
4	Variable names shall use the snake_case naming convention.	
5	The system shall display a 10-year savings projection using a loop.	
6	The program shall respond to all inputs within 2 seconds.	

Explain the difference between functional and non-functional requirements (1–2 sentences):

C3. IPO chart 3 marks

A program calculates the cost of installing solar panels. The user enters the **number of panels** and **cost per panel**. The program multiplies them for the subtotal, adds 10% GST, and displays the total. Complete the IPO chart:

Input	Processing	Output

C4. Data dictionary 5 marks

For the same scenario as C3, create a data dictionary with **at least 5 variables**. Use snake_case.

Variable name	Data type	Scope	Description

C5. Reading an SRS 3 marks

SRS — Solar Cost Calculator
FR1: Accept quarterly electricity bill as a float. **FR2:** Calculate annual bill as quarterly bill × 4. **FR3:** Use if/elif/else to determine tier pricing.
NFR1: All variable names use snake_case. **NFR2:** Include comments explaining calculation logic.
Constraint: Must run in a standard Python 3 terminal (no GUI).
Out of scope: Battery storage calculations.

(a)	What data type must the quarterly bill be stored as, and why?
(b)	What is the purpose of the "Out of scope" section?
(c)	How is a **constraint** different from a non-functional requirement?

Part D: From Design to Code /10

D1. Pseudocode to Python 4 marks

Translate this pseudocode into working Python code. Include # comments explaining your logic.

```
INPUT roof_space AS float
system_kw ← roof_space × 0.15

IF system_kw > 10 THEN
    category ← "commercial"
    rate ← 1100
ELSE IF system_kw > 5 THEN
    category ← "standard"
    rate ← 1300
ELSE
    category ← "small"
    rate ← 1500
END IF

total_cost ← system_kw × rate
DISPLAY category, total_cost
```

D2. Testing table 3 marks

Write **three** test cases for the pseudocode above. Choose inputs that test **different paths** through the if/elif/else.

#	Description	Input	Expected	Actual	P/F	Remedy
1						
2						
3						

D3. Factors of an SRS 3 marks

An SRS can include **factors** that influence the solution design. For each factor below, give a **one-sentence example** relevant to building a Solar Cost Calculator app for a school.

Technical factor	
Economic factor	
Social/ethical factor	

Marks summary

Part A: Read the Code	/10
Part B: Write the Code	/12
Part C: SD Theory	/18
Part D: Design to Code	/10
Total	**/50**

Checklist

- ☐ Code outputs traced by hand (Part A)
- ☐ Written code is complete Python (Part B)
- ☐ F/NF requirements identified (C2)
- ☐ IPO chart has all three columns filled (C3)
- ☐ Data dictionary uses snake_case, correct types (C4)
- ☐ Testing table covers different code paths (D2)