مكتب

Homework Assignment

Week 6: Planning Before Coding

Total Points: 100 — Due: Next Class

Name: ______________________ **Date:** ____________

Important Instructions

- Work on your own, but you may ask family members for hints only if stuck
- Write your code in Python *(using Thonny or any online IDE)*
- For setting up your code files, kindly see `instructions.pdf`
- For written answers, use the spaces provided on this sheet
- Show your work for full credit!

1 Part 1: Translation Practice (15 points)

Time estimate: 20 minutes

1.1 Exercise 1.1: Pseudocode to Python (8 points)

Convert the following pseudocode to Python code:

Pseudocode: Lunch Money Calculator

```
1. Ask user for their weekly lunch budget
2. Get the budget and convert to number
3. Calculate daily allowance (weekdays only)
4. Display the daily allowance amount
```

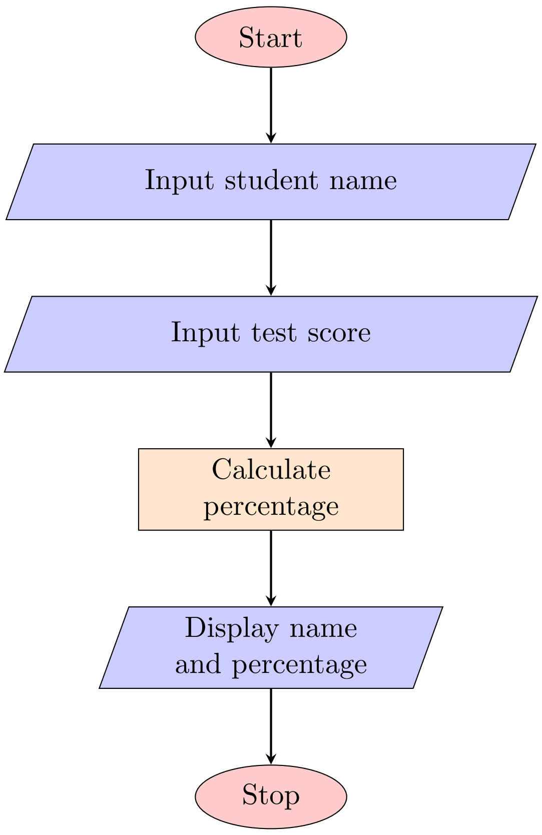
Open `week6_ex1_1.py` and write the Python code based on the pseudocode above:

```python
# Lunch Money Calculator (Partial Implementation)
# Write your Python translation here:
print("=== Lunch Money Calculator ===")
# Write your code here:
```

Save this code in `week6_ex1_1.py`

1.2 Exercise 1.2: Flowchart to Code (7 points)

Convert this flowchart to Python code:



Note: This flowchart only shows sequential steps (no decisions yet). Next week we'll add decision diamonds!
Open `week6_ex1_2.py` and write the Python code based on the flowchart above.

2 Part 2: Algorithm Design Practice (20 points)

Time estimate: 25 minutes

2.1 Exercise 2.1: Plan a Grade Message System (10 points)

Write pseudocode for a program that gives students feedback based on their score:

- 90 or above: "Excellent work!"
- 50-89: "Good job!"
- Below 50: "Let's work together to improve!"

Your Pseudocode

1. ____________________________
2. ____________________________
3. ____________________________
4. ____________________________
5. ____________________________
6. ____________________________

Now implement **only** the input and calculation parts in Python *(we'll add the decision logic next week)*. Open `week6_ex2_1.py` and complete the code:

```python
# Grade Feedback System (Input Only)
print("=== Grade Feedback System ===")
_____________________________     # Get student name
_____________________________     # Get test score
_____________________________     # Convert to number
_____________________________     # Calculate percentage
print(f"Thank you {___________}! Score recorded: {___________}")
print("Next week we'll add the feedback messages!")
```

Save this code in `week6_ex2_1.py`

2.2 Exercise 2.2: Design a Water Bill Calculator (10 points)

First, break down this problem into steps using pseudocode:

Problem: Calculate Monthly Water Bill

The water company charges:

- First 100 liters: Rs. 5 per liter
- Above 100 liters: Rs. 10 per liter

Create a program that calculates the total bill.

Flowchart: On a clean piece of paper, first draw a flowchart for this problem.

After you've drawn the flowchart, write pseudocode to outline the steps your program will take:

Your Pseudocode

```
1. ___________________________
2. ___________________________
3. if ___________________________
   (a) ___________________________
4. else
   (a) ___________________________
   (b) ___________________________
   (c) ___________________________
5. ___________________________
```

Note: We'll implement this program next week when we learn if statements in python!

3 Part 3: Code Ordering Challenge (15 points)

Time estimate: 20 minutes

3.1 Exercise 3.1: Fix the Recipe Order (8 points)

These lines make a program that converts cooking measurements. You task is to order the statements in the second column so they run correctly. The first column shows the order number that you need to fill in.

Order	Code Line
——	`print(f"{cups} cups = {tablespoons} tablespoons")`
——	`cups = float(input("Enter number of cups: "))`
line 1	`print("=== Kitchen Converter ===")`
——	`tablespoons = cups * 16`
——	`teaspoons = tablespoons * 3`
——	`print(f"That's also teaspoons teaspoons!")`

Open `week6_ex3_1.py` and write the complete ordered program.

3.2 Exercise 3.2: Sequence the Score Keeper (7 points)

Reorder the lines of this cricket score calculator program. The first column shows the order number that you need to fill in.

Order	Code Line
——	`total_runs = runs_from_ones + boundary_runs`
——	`sixes = int(input("Number of sixes: "))`
——	`boundary_runs = (fours * 4) + (sixes * 6)`
——	`print("Cricket Run Calculator")`
——	`runs_from_ones = int(input("Runs from singles: "))`
——	`print(f"Total runs scored: {total_runs}")`
——	`fours = int(input("Number of fours: "))`

Open `week6_ex3_2.py` and reorder the lines in the second column to make the program run correctly.

4 Part 4: Problem Decomposition Workshop (20 points)

Time estimate: 25 minutes

4.1 Exercise 4.1: Break Down the Birthday Party Planner (10 points)

Use the 5-step decomposition process to plan a program:

Problem: Birthday Party Cost Calculator

Create a program that helps plan a birthday party by calculating:

- Total cost of food *(based on number of guests and cost per person)*
- Decoration budget *(fixed amount)*
- Whether the total is within budget
- How much money is left over *(or how much more is needed)*

Step 1: Understand the Big Problem (What's the goal?) In one sentence, what should this program do?

- **Goal:** ______________________________

Step 2: Identify Major Tasks (What are the main things to do?)

1. ____
2. ____
3. ____
4. ____

Step 3: Break Down Each Task (What specific actions for each task?)

For Task 1, I need to:

- ____
- ____

For Task 2, I need to:

- ____
- ____

For Task 3, I need to:

- ____
- ____

For Task 4, I need to:

- ____
- ____

Example to help you:
If Task 1 was "Get party information", then:

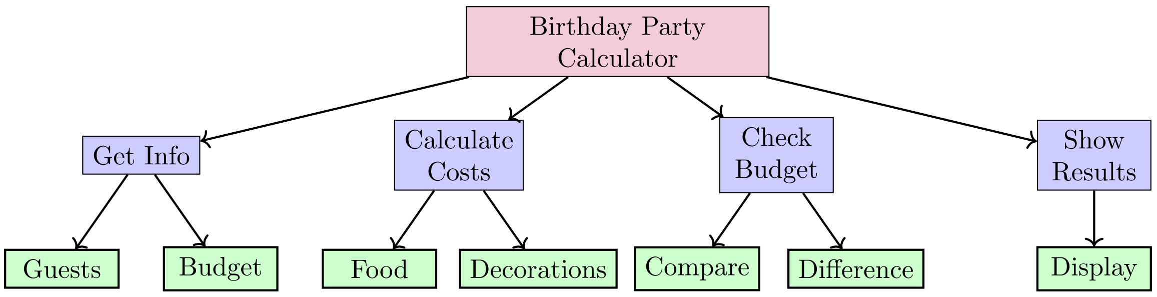
For Task 1, I need to:
- Ask how many guests are coming
- Ask what the budget is

Step 4: Order the Pieces *(What order makes sense?)*
Number your tasks from Step 2 in the order they should happen:

- Task ______ should be first because ___________________________
- Task ______ should be second because ___________________________
- Task ______ should be third because ___________________________
- Task ______ should be last because ___________________________

Step 5: Connect the Pieces *(How does information flow?)* Think about what information each task needs from other tasks. For example:

- To calculate food cost, I first need to know: ___________________________
- To check the budget, I first need to know: ___________________________



4.2 Exercise 4.2: Apply Your Decomposition (10 points)

Now write pseudocode based on your decomposition from above. Turn your tasks into numbered steps:

Party Planner Pseudocode

Based on your decomposition, write the steps:

1. ___________________________
2. ___________________________
3. ___________________________
4. ___________________________
5. ___________________________
6. ___________________________
7. ___________________________
8. ___________________________
9. ___________________________
10. ___________________________
11. ___________________________
12. ___________________________
13. ___________________________
14. ___________________________

Hint: Your 4 main tasks might become 8-10 steps when you include the specific actions!

5 Part 5: Creative Planning Challenge (25 points)

Time estimate: 30 minutes
Choose one of the following projects. Create detailed planning documents:

5.1 Option A: Sports Day Event Calculator

Plan a program that helps organize a school sports day:

Program Requirements

Your program should (when complete):

- Ask for number of students participating
- Ask for number of events
- Calculate total time needed (15 minutes per event)
- Determine if it fits in school hours (9 AM - 2 PM = 5 hours)
- Calculate how many water bottles needed (2 per student)
- Show the schedule

On clean pieces of paper draw both flowchart and pseudocode for the above program.

5.2 Option B: Recipe Ingredient Calculator

Plan a program that scales recipe ingredients:

Program Requirements

Your program should (when complete):

- Ask for original number of servings
- Ask for desired number of servings
- Get amounts for 3 ingredients
- Calculate scaled amounts
- Check if any amount exceeds 1000 (grams/ml)
- Warn if quantities are very large

On clean pieces of paper draw both flowchart and pseudocode for the above program.

6 Part 6: Reflection Questions (5 points)

Time estimate: 10 minutes
Answer these questions in complete sentences:

1. How did writing pseudocode first help (or not help) you write the actual Python code? Give a specific example from this homework.

2. Which planning tool do you find most helpful: pseudocode, flowcharts, or decomposition? Why?

3. Think of a real-world task you do (like getting ready for school). How would you decompose it into smaller steps?

7 Bonus Section: Extra Challenges (Optional - 10 extra points)

Only attempt if you've finished everything else!

7.1 Challenge 1: Complex Planning (5 points)

Create a detailed flowchart for a program that helps students track their daily screen tim. The program should:

- Ask for hours spent on: homework, gaming, social media, and videos
- Calculate total screen time
- Check if total exceeds 4 hours (recommended daily limit)
- Give different messages based on usage patterns
- Suggest breaks if any single activity exceeds 2 hours

On clean pieces of paper draw flowchart for the above program.

7.2 Challenge 2: Decomposition Master (5 points)

Use the 5-step decomposition process to break down this complex problem:

Problem: School Cafeteria Menu Planner

Create a program that helps the cafeteria plan weekly menus by:

- Tracking student preferences (votes for different dishes)
- Calculating ingredients needed based on expected students
- Checking nutritional balance
- Staying within budget
- Avoiding repeating popular dishes too often

Show your complete decomposition:

1. **Goal:** ____________________________
2. **Major Tasks:**
   - ____________________________
   - ____________________________

3. **Subtasks:** *(Break down at least 2 major tasks)*
4. **Task Order:** ____________________________
5. **Data Flow:** ____________________________

On clean pieces of paper write your complete decomposition. There will be many more steps then shown here, so use as many lines as you need!

Submission Checklist

Before submitting, make sure you have:

- ☐ Completed all required sections
- ☐ Tested each code file individually to ensure it runs without errors
- ☐ Drawn flowcharts, written pseudocode, decomposition steps where requested
- ☐ Answered all reflection questions
- ☐ Attempted bonus section *(optional)*
- ☐ Compressed your code files into a single zip file named `week6_hw.zip`

Time Tracking:
How long did this homework take you? _______________ hours

Parent/Guardian Signature: _______________________________
(Confirming student completed work independently)

Excellent work planning before coding!
Next week: Making Decisions with If Statements