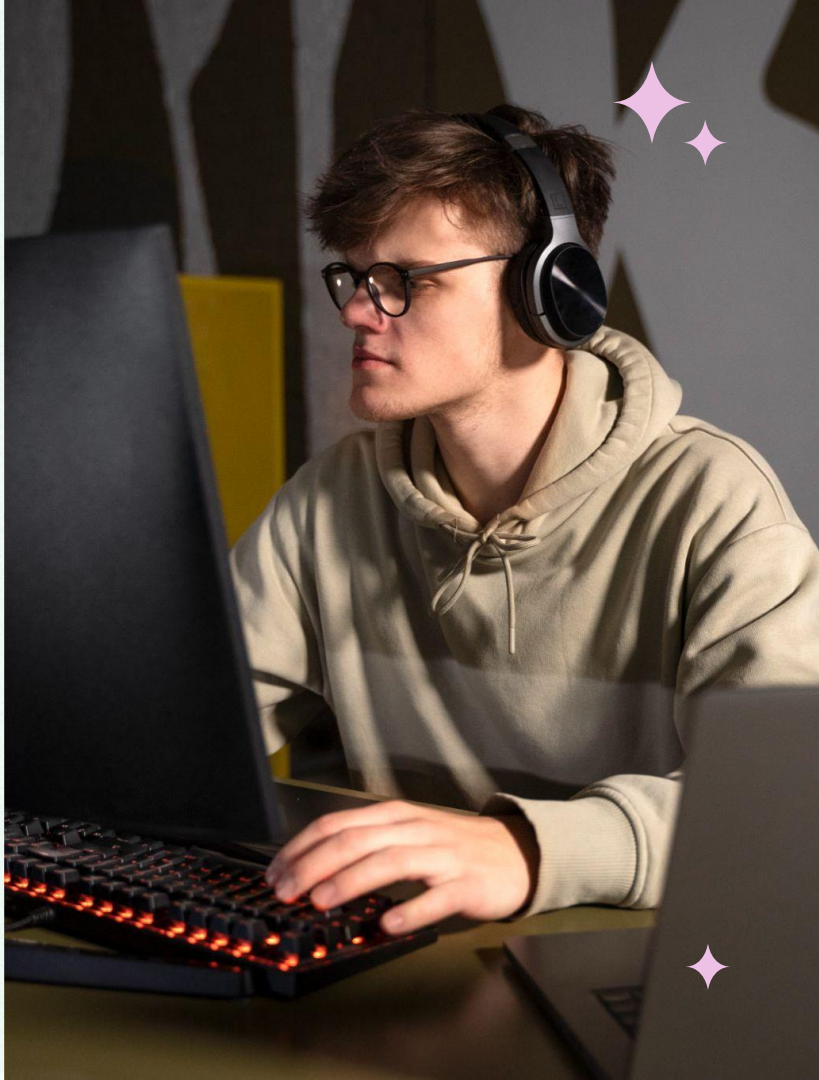
TOP-DOWN APPROACH

In this presentation, we'll explore what the top-down approach entails and discuss how to apply it to your assignments.

TABLE OF CONTENTS

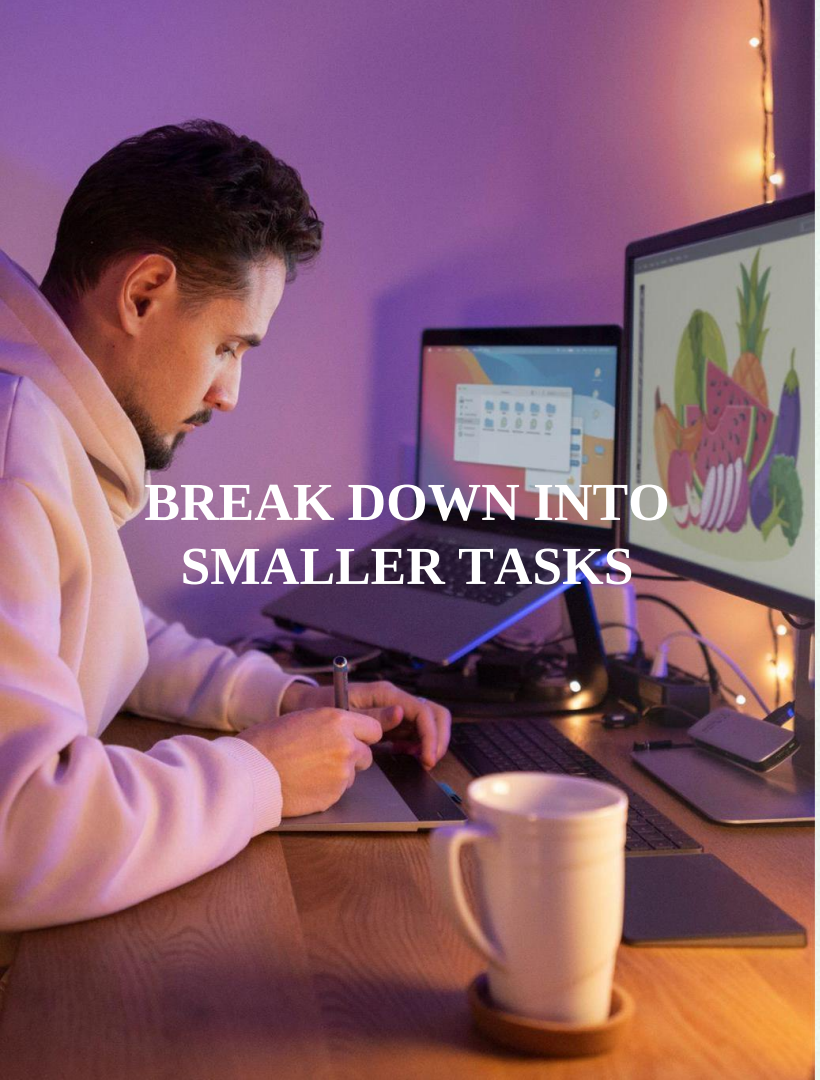
WHAT'S TOP-DOWN APPROACH

- The top-down approach is a problem-solving or design strategy that starts with a broad overview and gradually narrows down to the details.
- It's an effective method for tackling complex tasks by breaking them into manageable parts.
- It can be applied to various fields, including software development, project management, and even creative writing.



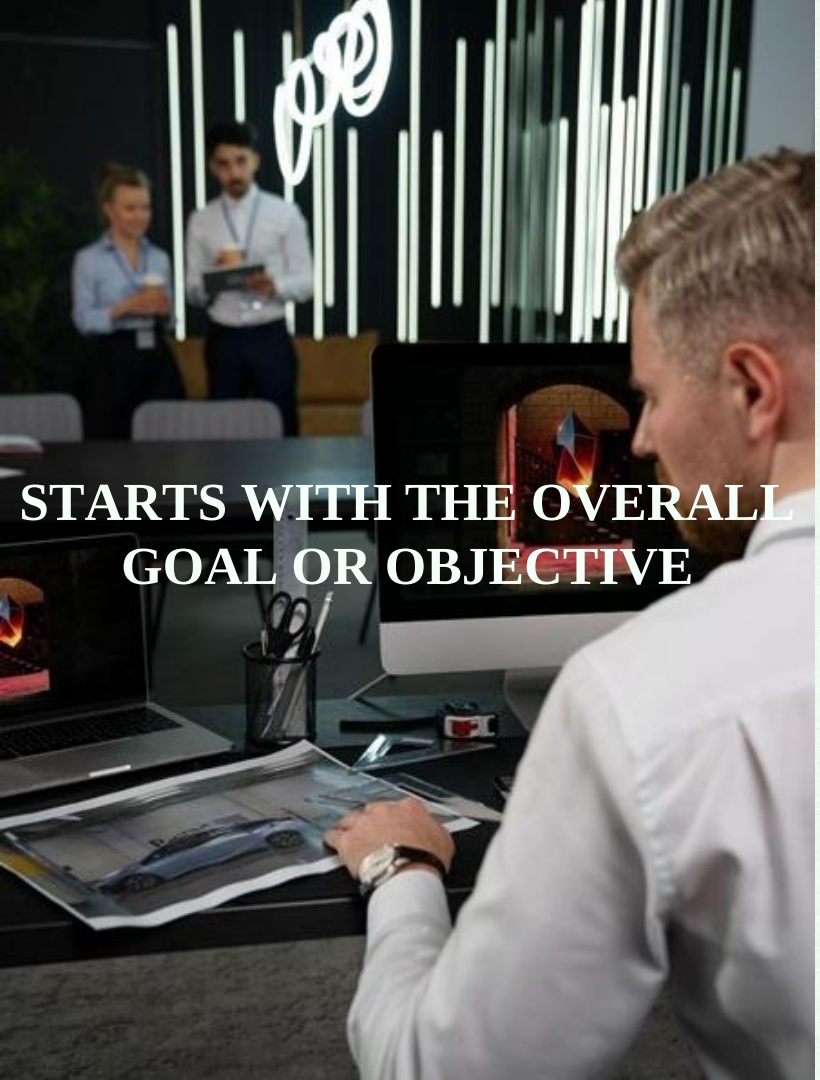
KEY CHARACTERISTICS

- Break down into smaller tasks
- Starts with the overall goal or objective
- Focus on structure and hierarchy
- Top-level guidance for lower levels



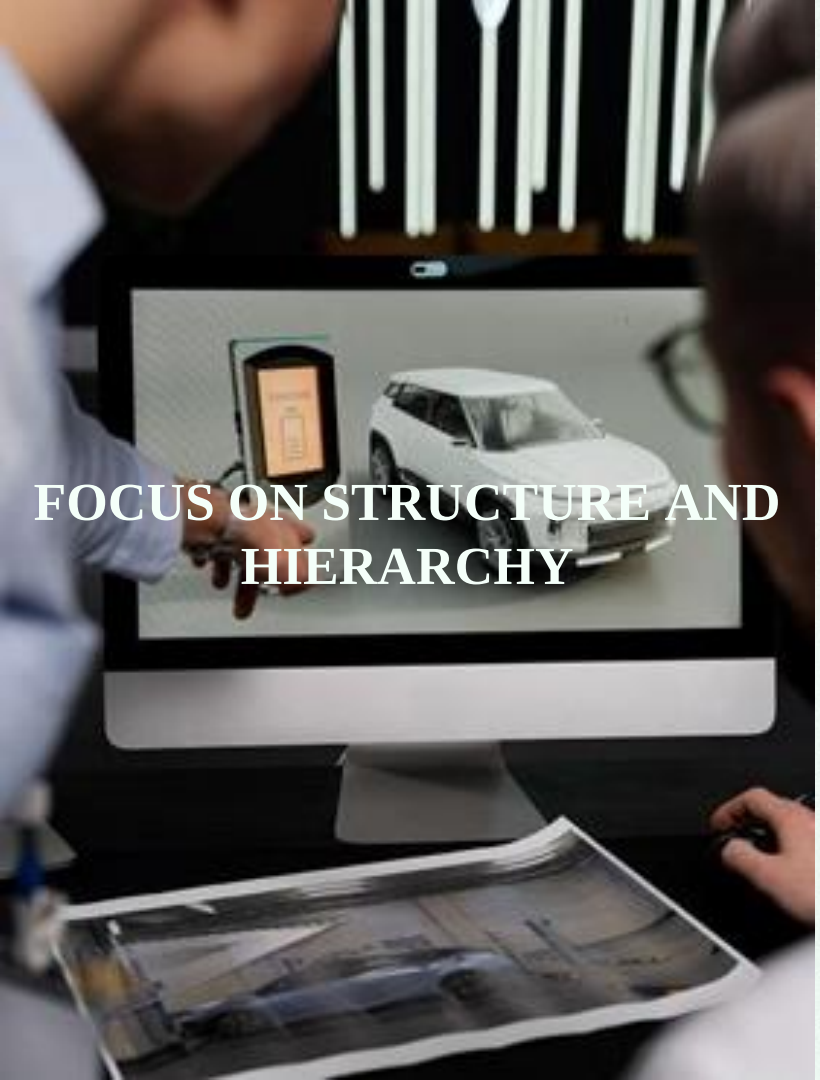
BREAK DOWN INTO SMALLER TASKS

- Before diving into details, the top-down approach focuses on clearly defining the desired outcome. This provides a guiding light for subsequent steps.



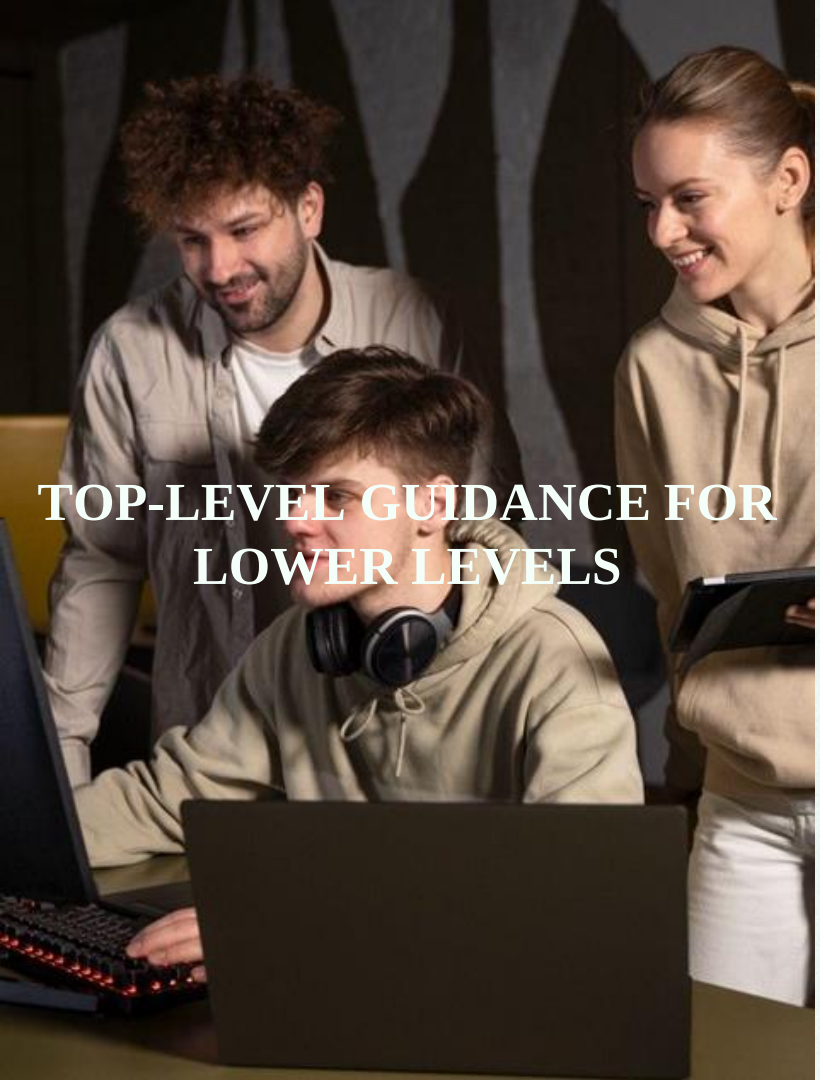
STARTS WITH THE OVERALL GOAL OR OBJECTIVE

- The overall goal is then decomposed into smaller, more manageable sub-tasks.
  This process can be further repeated until the tasks are at a level where they can be easily executed



FOCUS ON STRUCTURE AND HIERARCHY

- The top-down approach typically emphasizes a hierarchical structure, with larger goals at the top and smaller tasks branching down.
This provides a clear roadmap for progress.



TOP-LEVEL GUIDANCE FOR LOWER LEVELS

- Decisions and directives flow from the higher levels in the hierarchy down to the lower levels.
  This ensures alignment with the overall vision and strategy.

HOW THE TOP-DOWN APPROACH WORKS?

It generally refers to a way of solving a problem. Especially in computer science algorithms.

Step 1

Take the whole problem and split it into two or more parts.

Step 2

Find solutions to these parts.

Step 3

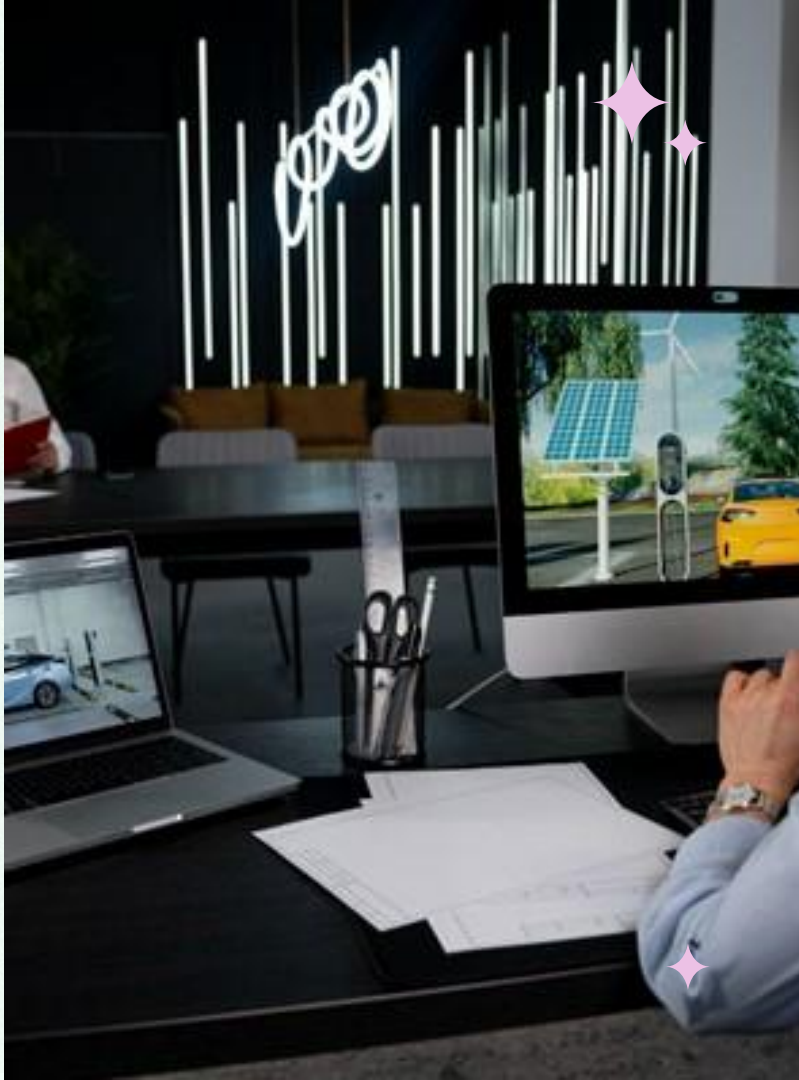
If these parts turn out to be too big to be solved as a whole, split them further and find solutions to those sub-parts.

Step 4

Merge solutions according to the sub-problem hierarchy thus created after all parts have been successfully solved.

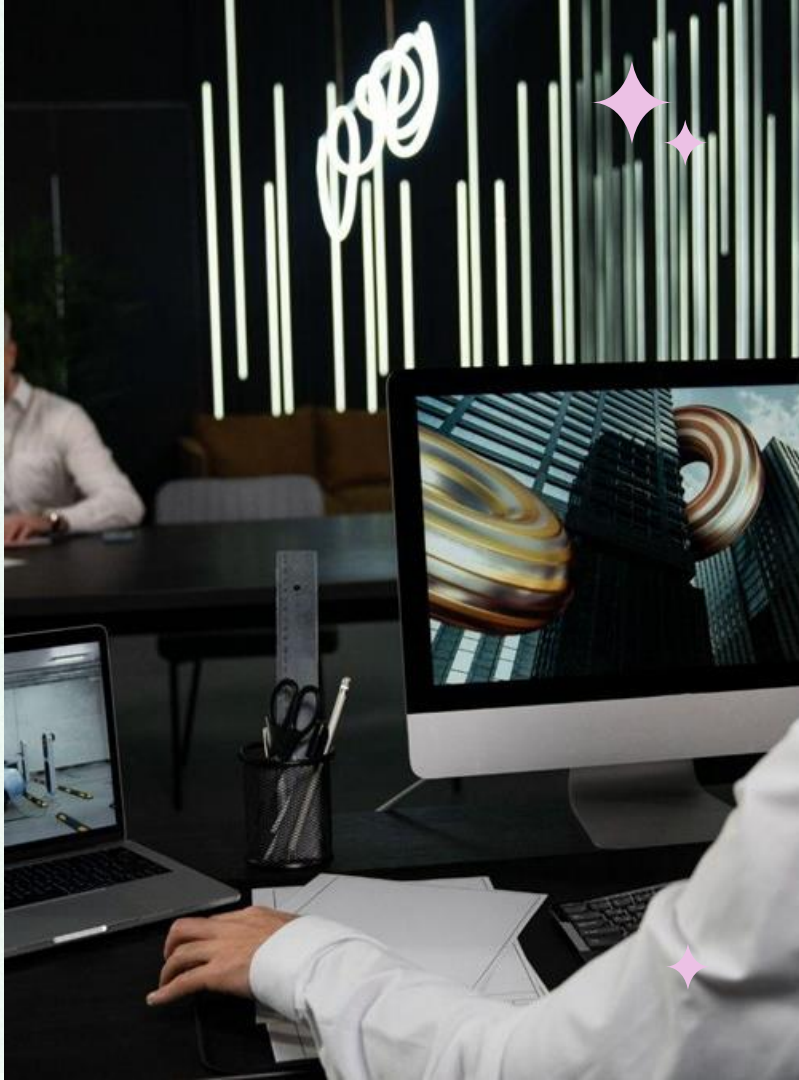
BENEFITS OF USING TOP-DOWN

- Clear goal orientation: Keeps you focused on the main objective.
- Organized structure: Makes the program easier to understand and debug.
- Efficient development: Breaks down complex problems into smaller, manageable steps.



THINGS TO CONSIDER

- Can be inflexible if requirements change later.
- May not capture all necessary details from the bottom up.
- Relies on good planning and decision-making at the top.



APPLYING TOP-DOWN TO THE PROGRAM ASSIGNMENT

To apply the top-down approach to the program assignment, I follow these steps:

Define the Overall Goal

- What is the main problem the assignment is trying to solve?
- What specific outcome do me want it to achieve?

Break Down the Goal

- Identify the main steps or processes needed to achieve the goal.
- Divide these steps into smaller, more concrete tasks.

Design the Subtasks

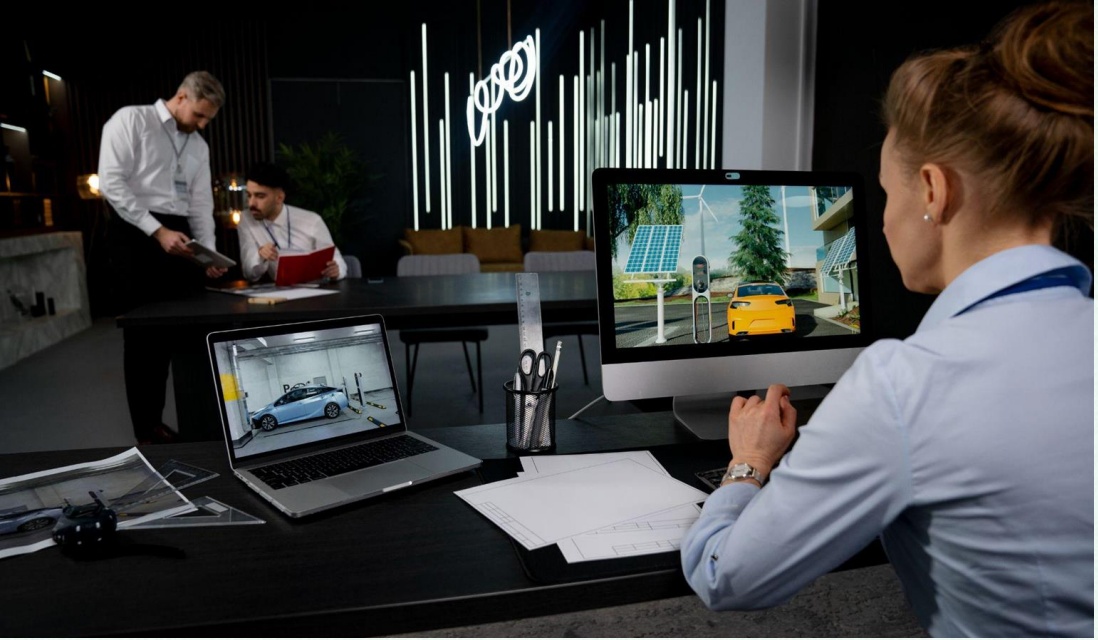
- For each subtask, determine what needs to be done and how it will be accomplished.
- Consider what inputs and outputs each subtask requires.

Assemble the Program

- Put the subtasks together in a logical order, forming the overall program structure.
- Implement each subtask using appropriate programming constructs and data structures.

Test and refine

- Thoroughly test your program at each level, from individual subtasks to the entire program.
- Identify and fix any bugs or errors that arise.



THANKS!