Program Design

ENGR 1181
MATLAB 01
Program Design in Real Life

Complex manufacturing processes require careful logic mapping to determine what actions should be taken and when to produce a product. Often in automotive manufacturing, the control of robots is carefully planned and timed with flowcharts at each stage of the welding process.
Today's Learning Objectives

- After today’s class, students will be able to:
  - Explain the significance of the planning stage in computer problem solving.
  - Create algorithms to plan and develop code.
  - Construct flow charts to plan and develop code.
Why Computer Problem Solving?

- Experience with open-ended problems
- Practice with problem solving
- Learning an important engineering skill
Problem Solving

1. Define
2. Represent
3. Plan
4. Implement
5. Evaluate
Algorithms

- Algorithms are a step by step plan
- Consider every step that’s necessary to complete a task or solve a problem
- Useful tools for problem solving (DR. PIE!)
- They will make programming easier and less time consuming 😊
Algorithm Exercise

- You will work with your classmates to finalize a step-by-step procedure to solve a problem or complete a task.

- Discuss amongst your teammates the various algorithms each person created.

- Select the algorithm that will best complete the task and modify the algorithm if needed.
  - Make sure you consider every small step or detail
  - You will need to make many assumptions
The Task: Have a person fill a bottle with stones.

- Start by listing assumptions
  - A bottle is present
  - Stones are present
  - ...
  - ...
  - ...

Algorithm Example
Algorithm Example

- Trade the finalized algorithm with another team
- Follow their directions EXACTLY
- Document how well it works!
Algorithm Example

- How did things go?
- Anything left out or overlooked?
  - Assumptions
  - Steps
  - Order
Algorithm Example

1. Set bottle upright near stones.
2. If bottle is closed, open it.
3. If bottle is full, go to #7.
4. Select a stone and try to place it in the bottle.
5. If stone is too large, discard stone and go to #3.
6. Place stone in bottle and go to #3.
7. Stop.
Structured Problem Solving

- Top-down, step-wise refinement
  - Start big.
  - Determine subtasks and order
  - Continue until you can’t get any smaller

- Pseudo Code
  - Informal
  - Programming “light”
Structured Problem Solving

- Flowcharts
  - Develop, not document
  - High-level
  - Executable only
  - Coding guideline
  - Specifics not included

- Symbols
- Combine with algorithm
Flowchart Example

- Flowcharts help us visualize our algorithm/program
- DR. PIE: Represent
- It’s good practice to make a flowchart before writing any program. It will save you lots of time!
Flowchart Example

- Friendship Algorithm
Flowchart Example

THE FRIENDSHIP ALGORITHM
DR. SHELDON COOPER, PH.D.
Flowchart Example

1. Start
2. Any Stones left?
   - Yes: Get Stone
     - Does Stone Fit?
       - No: Discard Stone
       - Yes: Put Stone in Bottle
         - Is Bottle Full?
           - No: Go back to Any Stones left?
           - Yes: End
   - No: End
Important Takeaways

- Programs require logic and information
  - Flowcharts map this flow and set the foundation for an efficient program
- Machines don’t know anything unless they are told exactly what’s needed
  - Think like a machine when programming!
  - Ex: Machines don’t know a bottle is upright.
What’s Next?

- Start working on MAT-01 homework.

- Introduction to MATLAB
  - Students will be exposed to the basic set-up and functionality of MATLAB.
  - Complete the pre-class reading and take the Carmen quiz before coming to class.