



Overview

Tweaking the words of Benjamin Franklin, an ounce of preparation is worth a pound of cure. Syntax and the concepts of programming languages are the structure of your coding projects. However, planning how to use those concepts to make your program work is critical to building a solid foundation.

In this activity, you will be tasked with creating pseudocode *and* a flowchart for the operation of an electric coffee maker with specific features. Then you will write a reflection paper explaining which tool you prefer and why.

Prompt

Your objective is to plan the program flow and logic for an electric coffee maker with the following features:

1. A clock displaying the current time. To set the time, simply press the ON button twice, then read the hour and minute from the user. When ON is pressed one more time, the time is set.
2. A pre-set time (hh-mm) to start brewing the coffee
3. Three buttons labelled OFF, ON, and PROGRAM:
 - a. When the OFF button is pressed, the machine does nothing other than display the current time.
 - b. When the ON button is pressed, the machine starts brewing coffee.
 - c. When the PROGRAM button is pressed, the machine waits for the pre-set time to begin brewing.
4. An automatic sensor that detects whether all the water has been used. Once the water runs out, brewing stops.
5. A heating sensor that keeps the coffee at a fixed temperature as long as the button is set to ON or PROGRAM

The machine's operation can be described as follows. First, the user adds coffee and water to the machine. If the user presses the ON button, the system begins boiling the water and then mixing it with coffee. The mix goes through the filter into the glass container. If the OFF button is pressed, the machine shuts down and the current time is displayed. If the PROGRAM button is pressed, the machine goes into sleep mode until the preset time is reached. Once it's reached, the machine "wakes up" and makes coffee.

Directions

Remember that pseudocode and flowcharts should express ideas and concepts in English. They are intended to be read by human beings, not compilers. There are several free options you can use to create your flowchart, such as [Draw.io](https://draw.io), [Lucidchart](https://lucidchart.com), or [Creately](https://creately.com). Make sure to use a tool designed for creating flowcharts to ensure that you are using the appropriate symbols and shapes.

1. Consider the high-level architecture of your program and break it into categories like logic, user interaction, etc. Then think about how you might implement each category. Be sure that your **solution** takes into account all features and requirements.
2. Create the **pseudocode** for your program. Be sure to do the following:
 - Use control structures.
 - Use indentation and white space.
 - Keep it simple and concise.
3. Create a **flowchart** for your program. Be sure to do the following:
 - Use appropriate design elements such as start and end points, decision branches, and so on.
 - Use labels for all flowchart shapes and arrows.
 - Keep everything on one page for better readability.

4. **Reflect** on your design so far by reviewing your pseudocode and flowchart. Write a response that addresses the following questions:

- Does your program flow in a logical order?
- What variables are implied by your design? Do they account for everything in the problem statement?
- What sections of the code might make sense to put in main()? What pieces of the code would make sense in a function or functions?
- Which method do you prefer, the visual flowchart or the text-based pseudocode? Why?

Guidelines for Submission

Attach your pseudocode, flowchart, and reflection response (at least two paragraphs) to the assignment submission page. Your pseudocode and reflection paper should be Word files. Your flowchart should be created using one of the free design tools specified above, then exported as a PDF.

Module Three Assignment Rubric

Criteria	Complete (100%)	Proficient (85%)	Not Complete (0%)	Value
Functionality	Solution meets guidelines by addressing all features and requirements	Meets some "Proficient" criteria, but with errors or omissions	Does not attempt criterion	20
Pseudocode	Correctly captures the logic and flow of solution, including indentations, and so on	Meets some "Proficient" criteria, but with errors or omissions such as functionality of solution, lack of white space, indentations, etc.	Does not attempt criterion	30
Flowchart	Correctly captures the logic and flow of solution with proper flowchart symbols and labels	Meets some "Proficient" criteria, but with errors or omissions such as functionality of solution, incorrect flowchart symbols, or missing labels	Does not attempt criterion	30
Reflection Paper	Explains which design tool is preferred and why	Meets some "Proficient" criteria, but with errors, omissions, or inadequate length	Does not attempt criterion	20
Total:				100%