Zachary Meisner

11/11/2021

CS260

Module 3 Coffee Maker

- Does your program flow in a logical order?

Yes, my program flows in a logical order. I believe so because I stated all the functions and logic prior to the main function and then I called the functions in main. This means that each of the functions should be accessible when called upon and none will interact with one another unless they also are called within those functions. I even added an output to ask the user to add water in case there was not any and put that there should be a mark on the interface of the command console, so the user knows that a time is set.

- What variables are implied by your design? Do they account for everything in the problem statement?

I do not believe I announced all the variables in my pseudocode but announced the ones that are more unique. For example, if I were to announce every variable, I would have to take into consideration how we get actual time, assign the variables from that to a more user-friendly variable in my program, reformat the variables to output so we could get the 00:00:00 to keep everything looking clean etc. There are a lot of variables to take into consideration with many other parts of the code as well, especially for the interaction between the hardware in a sense of heating water up and outputting the coffee after it is brewed.

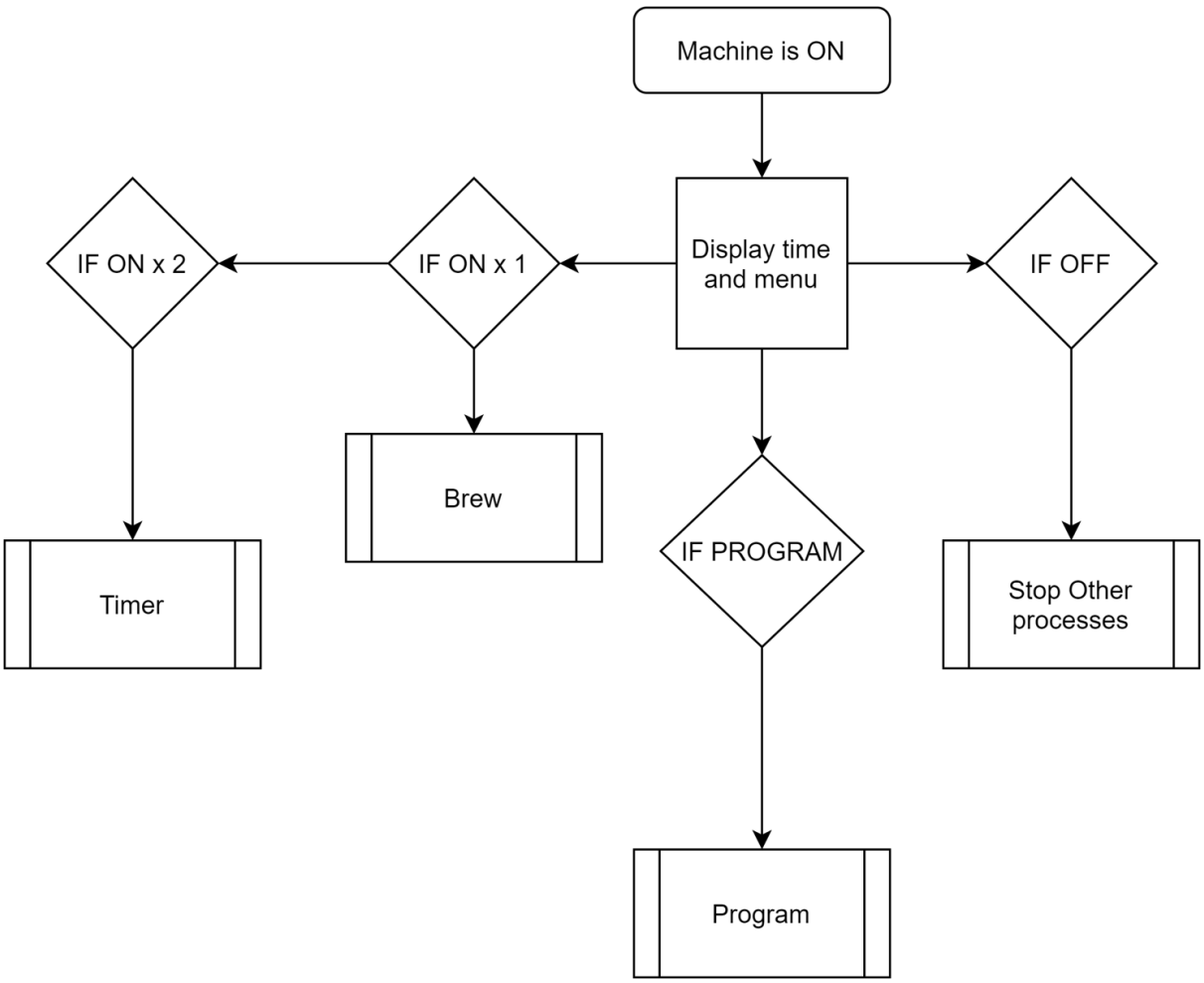
- What sections of the code might make sense to put in main()? What pieces of the code would make sense in a function or functions?

I do not think it is necessary to put all the functions in main because main only has to do three things. What main does is call three functions, now, based off of what is programmed in these functions is more important, because if we were to declare functions prior to main, and then layer them inside of one another it would be easier, especially if the project was object oriented and able to call any number of functions to work with one another opposed to a more structural design. This would allow the project to be updated more easily, and even make the coffee machine more efficient.

- Which method do you prefer, the visual flowchart or the text-based pseudocode? Why?

I think that text-based pseudocode is messier, and it is harder to implement but I think it is just as important as a visual flowchart. You can add more to the text-based pseudocode than you are able to a flow chart. Each are tools with their own purpose. In the case I had wanted to present

my application or my coffee maker to a crowd of people or shareholders in a company, the visual chart maybe better because it is better to present with. Overall, I prefer the visual flowchart because it is easier to read, but I think the importance of text-based pseudocode again points to the reason as to why we have manuals that come with the item, so users can really understand how it works without explaining all the code behind it.



Pseudocode

1. When machine on call main function

   ON = 1

   Important to clarify ON state for functionality of program

2. Basic Time logic for clock

   IF second > 59

   Minute += 1

   IF minute > 59

   Hour += 1

   IF hour > 12

   second minute hour = 0

3. Clarifies AM and PM

   IF hour <= 11

   AM

   ELSE

   PM

4. Brew Function

   INT Brew = 1;

   Sense water

   IF no water

   output Need Water!

Else

boil water

mix water with coffee

filter into container

Sense heat

user input OFF

Brew = 0;

5. Timer

Timer == 1; Distinguishes function to make sure proper input is taken

Hour = USER INPUT stores input

Minute = USER INPUT stores input

IF ON x1 if on is pressed again

Set Time = 1; save Hour and Minute in pre-set function of SET TIME

Timer == 0; End original functionality state

Main(); Return to main and display current time

6. Program

Sleep(); Wait for user time

IF SET TIME == TIME check current time and user time to see if they match

Brew(); If they match start brewing coffee

Else If they do not match keep outputting regular time

Add Checkmark on screen so user knows the time is preset

Main();

7. Call main and implement loop to update time and switch case to allow functionality

Main()

WHILE ON != 0 While machine has power

WHILE no button pressed

clear update and display current time

second ++ to make sure clock is ticking

sleep() for proper wait time between seconds

main() to continue loop while no button is pressed

Take user input

switch user input What happens when each button is pressed

case 1: ON

IF ON x1 If on button is pressed once

Brew(); call brew function

IF ON x2 if on button is pressed twice

Timer(); call timer function

case 2: PROGRAM

Program(); call program function

case 3: OFF

stop all current functionality except for WHILE ON loop