Daily Algorithms
Lab 02

Prepared by Matt Bennett
for ELEC 3150
Professor Carpenter - Fall 2016

I - Introduction

The purpose of this lab is to understand the importance of planning in the development of a program. Planning in this lab takes the form of both generalized pseudocode describing the programs decision making process and flowcharts that provide a detailed outline of the logical flow of the program.

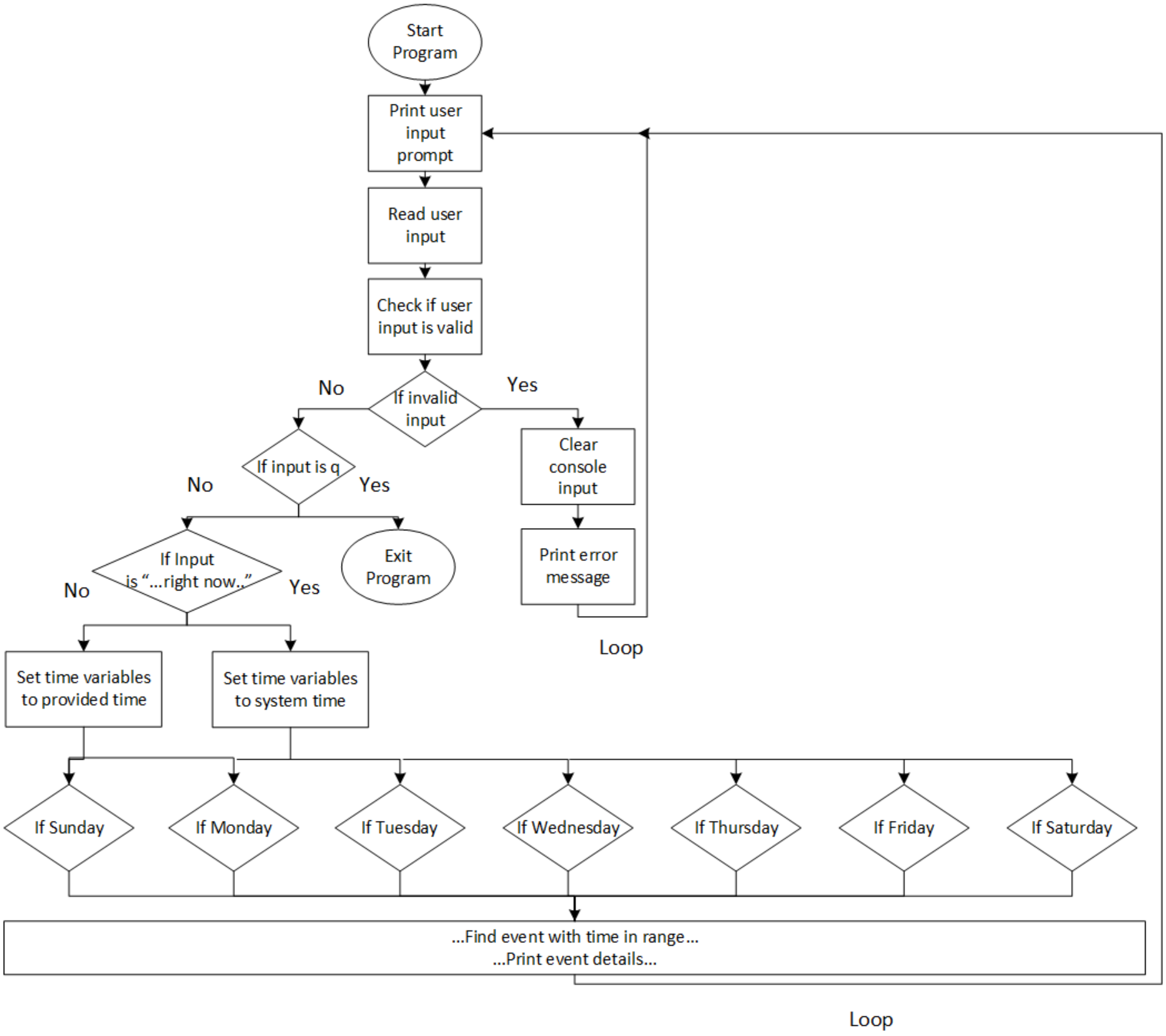
II - Procedure and Assumptions

Part 1 - Planning

- Prompt the user to provide a date and time in a loop, either by entering a time in the format, Weekday HH:MM xm, the phrase, “What am I doing right now?”, or “q” to quit.
    - Check for invalid data entry. Invalid data includes, mispellings, impossible or misformatted times, and times entries missing a weekday.
        - If invalid data is entered
            - Console buffer should be cleared to prevent future errors.
            - Provide a error message giving the required user input format.
            - Return to user entry prompt
    - If input is “q”, exit the program.
    - If input is “What am I doing right now?”, set time variables to system time.
    - If input is day/time formatted, set time variables to cooresponding day and time.
    - In the case of the particular weekday entered
        - In the case of time equal to or between an events boundary times
            - Print the event infomation to the console.
            - Return to the initial user menu prompt.
            - Determine and lookup the schedule data for the day of the week in question. The event in which the specified time occurs should then be determined and used in a response statement provided to the user.
                - Once the programs response has been printed to the console, the program should return to the initial prompt menu.

Assumptions:

- Each minute of the week coorresponds to one and only one event because logicially a person exists in one place at a time and only one place at a time.



Part 2 - Implementation

Implementation does not currently include menu for "What am I doing right now?" Additionally, program is not currently able to determine instances where a user has entered a string of text without spaces. For example, if a user enters "Monday08:30pm", the program will not continue until additional text is recieved followed by a carriage return.

In the algorithm the switch case of week days was replaced with a series of if/else statements in order to better account for events that are time but not day dependent.

III - Results

- Tested the determination of valid and invalid input for the following cases

  - ✓ Incorrent spelling or capitalization of week day
    - Mondai 08:20pm
    - thursday 04:14pm
  - ✓ Non-existent hours and minutes
    - Tuesday 13:06am
    - Friday 12:89pm
  - ✗ Single word entry
    - Saturday
    - 11:25pm
  - ✓ Correct date time format
    - Wednesday 08:45pm
  - ✓ Quit menu entry
    - q
- Single word entry input remains unresolved.

IV - Analysis

The program did not turn out as expected because in the psuedocode and flow chart planning phase C++ handling of multi-word space seperated string input was not considered. Accounting for this after initial planning proved to be challenging and future research is need on my part to better understand the string and cin C++ implementations and the best practices for multiword input checking.

V - Conclusions

The program correctly reads user input in cases where two words are entered in the console. The program cannot properly account for multiple word input, including the specification to accept user input, “What am I doing right now?” Planning program behavior prior to writing code was helpful as useful areas to implement functions were more apparent. Planning and documenting plans while still having a limited familiarity with C++, however, made the process more cumbersome than I generally would like. My personal preference in terms of coding indivually would be to have a rough plan and layout of functions and structures in my own personal format as opposed to the format of psuedocode and flowcharts. In the past, I have found the major benefit of these planning systems to be the ability to provide a general idea of code to others when colaborating on projects.