Psuedo Code for ZellersCongruence

1. Declare the constants for the following formula: h = (q + (26 \* (m + 1)) / 10.0) + k + (k / 4.0) + (j / 4.0) + (\*5 \* j) % 7 where q is the day of the month, m is the month, k is the year of the century, and j is the century
   * FIRST\_NUMBER set to 26
   * SECOND\_NUMBER set to 1
   * THIRD\_NUMBER set to 10.0
   * FOURTH\_NUMBER set to 4.0
   * FIFTH\_NUMBER set to FOURTH\_NUMBER which is 4.0
   * SIXTH\_NUMBER set to 5
   * SEVENTH\_NUMBER set to 7
   * YEARS\_IN\_CENTURY set to 100.0 for computing the year in century and century number
   * HEADER for the header asterisks since it’s used often
2. Declare computation and input variables
   * dayOfWeek\_Result to store the exact day of the week from the equation (i.e. 1-7)
   * dayOfWeek to store the actual day from the result (i.e. Monday, Tuesday, Wednesday, etc)
   * dayOfMonth to store input from user (1-30)
   * month to store the month from the user (1-12, but later computed as 3-14)
   * century to store the current century
   * yearOfCentury to store the current year in a century
   * year to store the user’s input
   * input as a Scanner to get the user’s input
3. Print the header in the format: \*\*\*\*\*\*\*\*\*\*Zeller’s Congruence\*\*\*\*\*\*\*\*\*\*
4. Prompt the user to enter a year, provide an example like 2020
5. Record the user’s input in year
6. Prompt the user to enter a month, make sure to tell the user to put it in the format of 1-12
7. Record the user’s input in month
8. Check if the user selected 1 or 2. If they did, then add 12 to their total months, making 1 (January) into 13, and 2 (February) into 14
9. Prompt the user for the day of the month, in 1-31 format
10. Close the input to reserve memory
11. Calculate the current century by doing the year divided by YEARS\_IN\_CENTURY (100.0), record in century
12. Calculate the current year of the century by getting the remainder (modulus) of the year divided by YEARS\_IN\_CENTURY (100.0), record in yearsOfCentury
13. Calculate the day by plugging the information into the formula, record in dayOfWeek\_Result:
    * (dayOfMonth + ((FIRST\_NUMBER \* (month + SECOND\_NUMBER)) / THIRD\_NUMBER) + yearOfCentury + ((century / FIFTH\_NUMBER) + (SIXTH\_NUMBER \* century)) % SEVENTH\_NUMBER
14. Store in variable dayOfWeek the day of the week, depending on what the result of dayOfWeek\_Result was:
    * 0 for Saturday, 1 for Sunday, 2 for Monday, 3 for Tuesday, 4 for Wednesday, 5 for Thursday, 6 for Friday
    * If for whatever reason the dayOfWeek\_Result is not within 1-7, make dayOfWeek null and display an error messaage
15. Display day of week if the variable dayOfWeek is not null
16. Show my name in the following format:
    * \*\*\*\*\*\*\*\*\*\*By Michael Amaya\*\*\*\*\*\*\*\*\*\*

ZellersCongruence Flow Chart:

A close up of a map

Description automatically generated

A close up of a map

Description automatically generated

UML Class Diagram:

A screenshot of a cell phone

Description automatically generated

Test Case:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cases | Input | Expected Output | Actual Output | Did it pass? |
| Case 1 | Enter year (e.g. 2020): 2017  Enter month (1-12): 1  Enter the day of the month (1-31): 27 | Day of the week is Friday | Day of the week is Friday | Y |
| Case 2 | Enter year (e.g. 2020): 2010  Enter month (1-12): 6  Enter the day of the month (1-31): 10 | Day of the week is Thursday | Day of the week is Thursday | Y |
| Case 3 | Enter year (e.g. 2020): 2015  Enter month (1-12): 5  Enter the day of the month (1-31): 16 | Day of the week is Saturday | Day of the week is Saturday | Y |
| Case 4 | Enter year (e.g. 2020): 2018  Enter month (1-12): 2  Enter the day of the month (1-31): 14 | Day of the week is Wednesday | Day of the week is Wednesday | Y |

Case 1 Screenshot:

A screenshot of a cell phone

Description automatically generated

Case 2 Screenshot:

A screenshot of a cell phone

Description automatically generated

Case 3 Screenshot:

A screenshot of a cell phone

Description automatically generated

Case 4 Screenshot:

A screenshot of a cell phone

Description automatically generated

Lessons learned:

In this lesson, I learned the importance making sure everything that needs to be in parenthesis is in parenthesis or the result will be wrong. I also learned the importance of making sure you divide by a double, or else you will receive an integer value. Furthermore, I learned that there are many ways to solve a solution. An example would be in the part where I add 12 to the month, you can also just change the month to 13 if the month is 1, and 14 if the month is 2, but it’s shorter to write month += 12.

Check List:

|  |  |  |  |
| --- | --- | --- | --- |
| **#** |  | **Y/N** | **Comments** |
|  | **Source java files** | **Y** |  |
|  | **Compressed files:** | **Y** |  |
|  | FirstInitialLastName\_Project2\_Moss.zip | **Y** |  |
|  | FirstInitialLastName\_Project2\_doc.zip | **Y** |  |
|  | **Program compiles** | **Y** |  |
|  | **Program runs** | **Y** |  |
|  | **Checklist is completed and included in the Documentation** | **Y** |  |
|  | **Documentation file:** | **Y** |  |
|  | **Comprehensive Test Plan** | **Y** |  |
|  | **Screenshots based on Test Plan** | **Y** |  |
|  | **UML Diagram** | **Y** |  |
|  | **Algorithms/Pseudocode** | **Y** |  |
|  | **Flowchart** | **Y** |  |
|  | **Lessons Learned** | **Y** |  |