Arman Khan

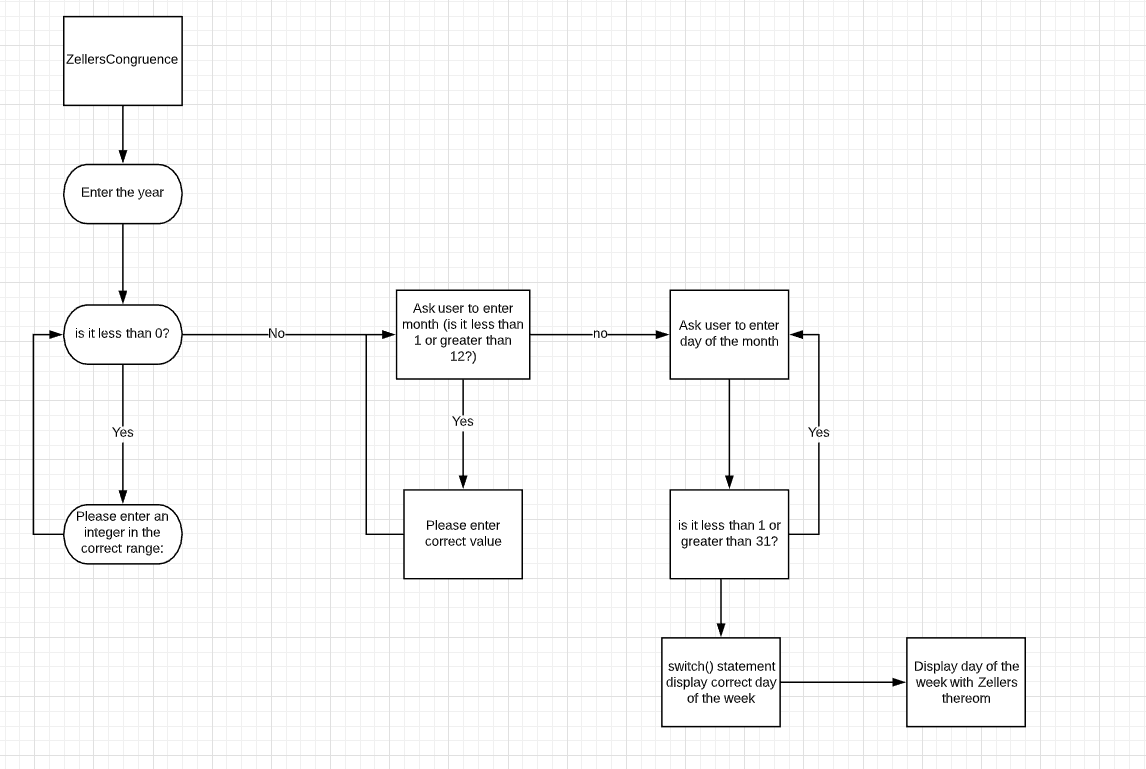
Pseudo code:

1. Create a scanner to read input
2. Declare variables listed in project
   1. Int h; day of the week
   2. Int q; day of the month
   3. Int y; year
   4. Int m // is month
   5. Int j; year/100
   6. Int k; year of century year %100
3. Prompt user to enter year
   1. Store answer in y
4. Prompt user to enter month
   1. Store answer in m
   2. Validate if month was in correct year using while loop
5. Prompt user to enter day of the month
   1. Store answer in q
   2. Validate if the day is in the correct month
6. If statement
   1. if month is January or February convert to 13 or 14 respectively
7. Zellers congruence formula stored in variable h
8. Switch statement depending on h’s value prints out specific day of the week

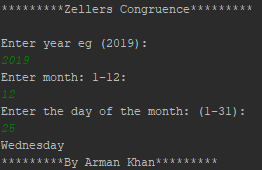
Test Plan (validation) :

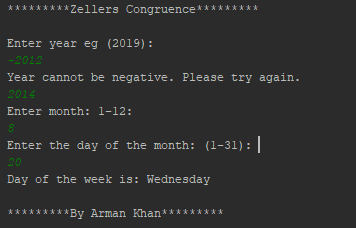
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cases | Input | Expected Output | Actual Output | Did Test Pass? |
| Case 1 | "Enter year eg (2019): " | User output is y (year) | year | Y |
| Case 2 | "Enter month: 1-12:"  -12 | "Please enter a valid number corresponding to month: " | "Please enter a valid number corresponding to month: " | Y |
| Case 3 | "Enter the day of the month: (1-31): "  0 | "Please enter a valid day (1-31)" | "Please enter a valid day (1-31)" | Y |

Flow Chart:



Test Code Screenshots:





UML class diagram:

|  |
| --- |
| **Class Name:**  **Main** |
| **+h: int**  **+q: int**  **+y: int**  **+m: int**  **+j: int**  **+k; int** |
| **+switch(h)** |

Lessons learned:

I learned to troubleshoot my code a little bit more and get used to differing syntax between programming languages. I also made sure to utilize input validation in all my code, where the user could have mistype or try to mess with the program. I also found out that this formula struggles with Leap years I believe.