

Project 4: Birthday Calculator

High Level Description:

The purpose of this project is to practice using conditional execution, Scanners, and cumulative algorithms to calculate the number of days until a user's birthday. The program will also print the "absolute day of the year" for both the current day, and the inputted birthday (the "absolute day of the year" is a date's place within the year from 1 to 365). You may assume for the entirety of this code that it IS NOT a leap year. Below are detailed instructions for how your program should function:



Program Function:

Your program should:

1. Prompt the user for today's date.
 - a. Each prompt should list the range of possible values from which to choose.
 - b. The range of possible values for the number of days should change based upon the month that the user entered (see example).
2. Print out the "absolute day of the year" for today's date.
3. Prompts the user for their birthday.
 - a. Again, the prompts should include a list of values as they did for today's date.
4. Print out the "absolute day of the year" for the user's birthday.
5. Print out the number of days until the user's birthday.
 - a. The program should print different messages if the user's birthday is near.

You may assume that all user input is valid (they will enter ints when prompted for an int). You may also assume that it IS NOT a leap year (February will have 28 days).

Here are 4 examples to help see how the program should function:

AP CS A - Unit 4 Project – Birthday Calculator

Run #1 This program tells you how many days it will be until your next birthday. Please enter today's date: What is the month (1-12)? <u>7</u> What is the day (1-31)? <u>24</u> 7/24 is day #205 of 365. Please enter your birthday: What is the month (1-12)? <u>11</u> What is the day (1-30)? <u>6</u> 11/6 is day #310 of 365. Your next birthday is in 105 days.	Run #2 This program tells you how many days it will be until your next birthday. Please enter today's date: What is the month (1-12)? <u>9</u> What is the day (1-30)? <u>22</u> 9/22 is day #265 of 365. Please enter your birthday: What is the month (1-12)? <u>3</u> What is the day (1-31)? <u>17</u> 3/17 is day #76 of 365. Your next birthday is in 176 days.
Run #3 This program tells you how many days it will be until your next birthday. Please enter today's date: What is the month (1-12)? <u>2</u> What is the day (1-28)? <u>14</u> 2/14 is day #45 of 365. Please enter your birthday: What is the month (1-12)? <u>2</u> What is the day (1-28)? <u>15</u> 2/15 is day #46 of 365. Wow, your birthday is tomorrow!	Run #4 This program tells you how many days it will be until your next birthday. Please enter today's date: What is the month (1-12)? <u>12</u> What is the day (1-31)? <u>25</u> 12/25 is day #359 of 365. Please enter your birthday: What is the month (1-12)? <u>12</u> What is the day (1-31)? <u>25</u> 12/25 is day #359 of 365. Happy birthday!

Grading:

Your grade on this project will be based around the following categories:

1. External Correctness (Output)

Does your code produce the exact desired result. When run, will your code prompt the user for inputs, then return the correct values for the “absolute day of the year” and for the number of days until the user’s birthday?

2. Internal Correctness (Application of Unit 4 Material)

Your code must include the required components from the chapter. Here are your specific constraints on your code:

- a. You must use at least 4 non-trivial methods in your program besides main.
 - i. Use procedural decomposition to break down your program into a series of methods that complete subtasks. You should use parameters and return values appropriately to pass information throughout your code.
 - ii. Each method should perform its own single, clear and coherent task. No method should do too much or carry too much weight. Aim for methods that are no greater than 20 lines in length.
 - iii. Non-trivial here means methods should be necessary and useful in your program. In other words, do not write a method simply to print a single line to the console, or convert an int to a double.
- b. You must also include at least 1 cumulative sum algorithm.
- c. Your main method should be limited to mainly calls to different methods executing subtasks. It should be a concise summary of the overall program. You should avoid “chaining” long sequences of method calls together without returning to main.
- d. You must use conditional execution with if/else statements. This is a key aspect of this unit, and a fundamental skill in programming. Make sure you do extra CYUs if you are struggling to use these properly.
- e. You may not use any material outside of class (and up to unit 4) for the base version of your code. If you would like to use Java features that we have not yet used in class for your extension, please check in with me first.

3. Coding Style

Your code must be neat, concise, and readable! Refer to the following coding conventions for this project:

AP CS A – Unit 4 Project – Birthday Calculator

- Name your class BirthdayCalculator and submit BirthdayCalculator.java to Teams by the assigned due date.
 - Include a block comment at the top of your program with your name, the period, the date, and a description of your program.
 - Use conventional indentation and whitespace.
 - Avoid lines that go over 100 characters in length - you can check this at the bottom of your Eclipse window.
 - Give meaningful names to all methods in your code.
 - Follow Java's naming standards about the format of ClassNames, methodNames and CONSTANT_NAMES.
 - Include a blank line followed by meaningful comments at the start of each method other than the main method, describing its behavior. This may be javadoc or regular block comments. This should not be how your method works, but an overview of what it is supposed to do.
 - Opening curly braces must be either at the end of the line beginning the block or on their own new line.
 - *For example:*
`public static void main(String[] args) {`
or
`public static void main(String[] args)`
`{`
 - Closing curly braces must be on their own line.
-

Project Breakdown:

Part 1 – Design:

Start by writing down the list of operations your program must perform and the order in which they must be completed. Use pseudocode, diagrams, and pictures to help you map out the steps you need to take to accomplish the task. Don't worry about being precise with the logic, just make sure you have a good idea of what you need your program to do.

Next, start dividing your program into methods that will each perform their own subtask. Find areas where certain behaviors are repeated multiple

times, or where the same behavior appears in multiple different places throughout the code. These will form your different methods.

Part 2 – Coding:

Create your BirthdayCalculator class and write the method headers for each of your different subtasks. Be sure to include meaningful names for each method and its parameters, and think about what inputs and outputs the method would use. Don't worry about filling in the details of the methods yet, just build the skeleton of your program by "stubbing" out each of your methods. Don't forget to add comments to each method!

Then, write each method one at a time. Use print statements to help you debug areas that aren't working as expected. Make sure you test each method frequently and think of different cases that might cause your program to function incorrectly. Once each method works on its own, start building out your main method to tie all your subroutines together to complete the desired result.

Part 3 – Extending:

To receive full credit, you must also find a way to personalize your code and challenge your abilities. This means adding extra complexity to your program to incorporate additional functionality. Use the same steps outlined in part 1 and part 2 to help you add your extension to your existing code. Go slow, plan ahead, ask questions, get creative, challenge yourself and HAVE FUN!

Here are a few ideas for you to get started with:

- Input validation – Assume the user does not always input their responses in the correct way. Use if/else statements, while loops, or try/catch statements to ensure that the user enters the correct datatype when prompted.
- Include Leap Years – Instead of assuming it is not a leap year, add functionality to detect a leap year and adjust your calculations accordingly.

AP CS A - Unit 4 Project – Birthday Calculator

- Get Precise – Add the exact number of hours, minutes, and seconds as well as the number of days until the user’s birthday.
- On this Day in History – Include a snippet of information about historical events that happened on the current day or on the user’s birthday (or both)! Here are a couple of websites that I have used for something similar:
 - <https://blogs.ams.org/mathgradblog/2012/09/07/day-mathematics-history/>
 - <https://pballew.blogspot.com/2024/11/on-this-day-in-math-november-14.html>
 - <https://www.history.com>this-day-in-history'>
- Get Astrological – Include information about the user’s astrological sign based on their birthday (horoscopes, traits associated with their sign, etc.). Add more prompts to get their birth year and time of birth to get more information on their different signs.
- Get Creative – If you have your own idea, let me know! I’d love to hear what you come up with 😊.

S. No.	Name of the Month	Number of Days
1.	JANUARY @ math-only-math.com	31
2.	FEBRUARY	28 Or 29
3.	MARCH	31
4.	APRIL @ math-only-math.com	30
5.	MAY	31
6.	JUNE	30
7.	JULY	31
8.	AUGUST @ math-only-math.com	31
9.	SEPTEMBER	30
10.	OCTOBER	31
11.	NOVEMBER	30
12.	DECEMBER	31 @ math-only-math.com