UNIVERSITY OF SOUTHERN QUEENSLAND
CSC1401 – Foundation Programming (Semester 3, 2022)
Assignment 1 Specification

Days-From-Birth

Due Date: 15th Dec 2022

Weight: 10%

Goals and Topics

The assignment problem is straightforward. All necessary details have been supplied. The solution to the problem will be straight line code which will use the programming concepts and strategies covered in Workshops 1-4. The subgoals are:

- Understanding values, variables, operations, and functions
- Translating simple design into Python code
- The mechanics of editing, interpreting, building, and running your program
- Testing your program
- Commenting your source code
- Becoming confident and comfortable with programming small problems

Your Task

"Days-From-Date" is an interesting and useful feature in website design. It shows up how many days (or years, hours, minutes, and seconds) after an event started. For example, a feature of Days-From-Birth shows the number of years, days, hours, minutes, and seconds after the date of birth. In this assignment, you are going to write a Python program to implement the "Days-From-Birth" feature.

The Concept of Time in Python

In Python, a time is defined as a **Date** Object. Each date object stores its state as a time value, which is a primitive number that encodes a date as seconds since 1 January 1970 00:00:00 UTC. Thus, a date later than 1 January 1970 00:00:00 UTC will have a positive time value, whereas an earlier date will have a negative time value. On the basis of the common timeline (which we all live on), the distance between any two dates can be calculated using their time values in seconds. Figure 1 illustrates the concept, where *C* is a date earlier than 1 January 1970 00:00:00 UTC, and *A* and *B* are later with *B* being further than *A*.

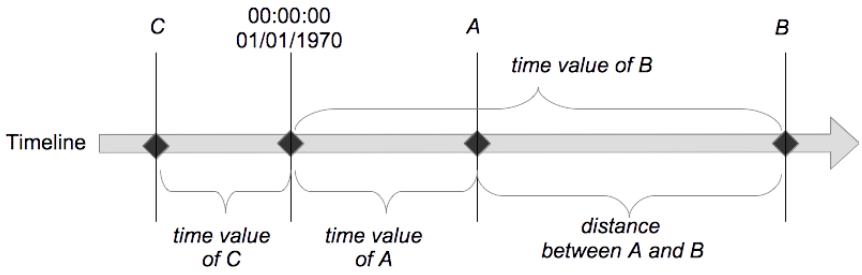

Figure 1: The Difference Between Two **Date** Instances

Functional Requirements

Variables

Within the script section, create variables following professional conventions and initialise them using the right values. Some variables have been suggested in the following tables. You should create more when necessary.

Table 1: Variables1

Description	Value
Number of seconds in a year#	365*60*60*24
Number of seconds in a day	60*60*24
Number of seconds in an hour	60*60
Number of seconds in a minute	60

In this assignment, we assume every year has 365 days.

Table 2: Variables2

Description	Initializing value description	Type
Year of the event	The year of event	*Number*
Month of the event	The month of event	*Number*
Day of the event	The day of event	*Number*

Calculation

1. Create a **Date** object for the birth by using the variables created previously.
    - Use three input functions to input the year, month, and day of the birth one by one with the hints "Please enter the year of your birth, e.g., 1970-2021", "Enter the month of your birth, e.g. 1-12" and "Enter the day of your birth, e.g. 1-31". No validation plan is required. We assume users will enter valid numbers according to hints. For example, if

the year of birth is not a leap year. The user won't enter 29 for the date of birth if the month is February. Then transfer them to the Date constructor.

Example of transferring number to the Date constructor and calculating how many seconds of the date since 1 January 1970 00:00:00 UTC:

```
>>> import datetime, time
>>> t = datetime.datetime(2011, 10, 21, 0, 0)
>>> time.mktime(t.timetuple())
1319148000.0
```

- Mind the order of arguments sent to the constructor

2. Obtain the current time instant, a floating-point number of seconds since "the epoch" (see Exercise 4.1).

3. Calculate the difference between the current time and the birth time **(assume it is the starting time 0:00:00 a.m. on the date of the birth):**
   - Get the birth time value in seconds as in the example above.
   - Deduct the birth time value by using the time value of the current time.
4. Calculate the number of years to the event:
   - Divide the time value difference by the number of seconds in a year (1 year =365 days).
   - Use the floor division to reduce the resulting number to an integer.
5. Calculate the number of days, hours, minutes, and seconds in the remaining time value:
   - Mod the time value difference by the number of seconds in a year.
   - Divide the mod result by the number of seconds in a day.
   - Use the floor division to reduce the number to an integer for the number of days.
   - Repeat the three steps above to calculate the number of hours, minutes, and seconds. You may need to update the calculating formula accordingly.

Presentation

Sample output is presented below when running the "Days-From-Birth" program.

"Your birthday is 25/7/1999 *(XXX years, XXX days, XXX hours, XXX minutes, and XXX seconds ago)."*

Note that

- the information should be displayed using the **print** function.
- wherever possible you should use variables in expressions instead of explicit values (e.g., numbers), The date format **must be** the same as "25/7/1999". Other formats (e.g., 25/07/1999 or 25 July 1999) are not acceptable; "XXX" is the result of your calculations.

Testing

Test your program by entering your birth date to see whether the outcome is correct.

Non-functional Requirements

Structure of the Source Code

- All code should appear in an *.ipynb* file.
- The marker will only test your code in the Jupyter notebook. Please make sure your submitted Jupiter notebook file is **readable** by Anaconda installed as the "Setting up Python on your computer" section. Students cannot use any libraries or built-in functions which require an extra python package installed.

Comments

- You are required to add at least three comments to the source code.
- Do not comment on every single line, instead, comment on blocks of code with a common purpose.
- Do not simply translate the syntax into English for comments, instead, describe the purpose of blocks of code.

Submission

What You Need to Submit – Two Files

For a complete submission, you need to submit two files as specified below.

The assignment submission system will accept only the files with extensions specified in this section.

1. *The program* in a file saved with an *.ipynb* extension contains the source code implemented following the functional and non-functional requirements.

2. *The program* in a file saved with a *.doc/.docx/.odt* extension contains the exact same source code in the *.ipynb* file. You can just download the *.ipynb* as the *.py* file, then copy the code in the *.py* file and paste it to the *.doc/.docx/.odt* file. If you don't submit your code in **both** *.doc/.docx/.odt* and *.ipynb* files, or *the code* in *.doc/.docx/.odt* and *.ipynb* files are **different**. You will get a penalty (up to 5 marks).

Marking Criteria

The assignment will be marked out of 10. Table 3 presents the marking criteria. If all criteria are satisfied, you will receive 10 marks. If not, all criteria are met, and part marks may be given. Check your own submission against these criteria before you submit.

Table 3: Marking Criteria

ID	**REQUIREMENTS**	MARK
	Functional Requirements	
1	The program is running without any syntax errors	0.5
2	The input function for entering the year is correct	0.5
3	The input function for entering the month is correct	0.5
4	The input function for entering the day is correct	0.5
5	Transferring inputs to the Date constructor correctly	0.5
6	The time of birth and current time instant, a floating-point number of seconds since "the epoch" is calculated correctly. The difference between the two times is calculated using an appropriate strategy.	0.5
7	The number of years from the event is calculated correctly	0.5
8	The number of days from the event is calculated correctly	1
9	The number of hours from the event is calculated correctly	1
10	The number of minutes from the event is calculated correctly	1
11	The number of seconds from the event is calculated correctly	1
12	The calculating result is displayed appropriately	0.5
13	The date format in the output meets the requirement	0.5
14	All the functions meet the requirements	0.5
	Subtotal	*9*
	Non-functional Requirements	
15	All variables are used appropriately, and identifiers of variables are following professional conventions. Variables are used in calculation and expression instead of explicit values	0.5
16	At least three comments are added to describe the purpose of blocks of code	0.5
	Subtotal	*1*
	TOTAL	**10**

Suggested Strategy

You have three weeks to finish the assignment. Plan to complete it on time. Do not write all of the code in one sitting and expect that everything will be working smoothly like magic.

First week Read assignment specification, clarify anything unclear by putting a post on the Assignment 1 Forum, think about how to do it, how to test it, and devise high-level algorithms for each independent part of the assignment. Begin to type the program (with comments), in incremental stages. Seek help on the assignment forum if needed.

Second week Re-read the specification, continue to refine the design of various sections to code, and bring up any problems to the assignment forum if necessary. Finish initial coding.

Third week Fully test the program; have another review on the source code; re-read the specification (especially marking criteria) to make sure you have done exactly what is required.

Plagiarism and Academic Misconduct

USQ has zero tolerance for academic misconduct including plagiarism and collusion. Plagiarism refers to the activity of presenting someone else's work as if you wrote it yourself. Collusion is a specific type of cheating that occurs when two or more students exceed a permitted level of collaboration on a piece of assessment. Identical layouts, identical mistakes, identical arguments, and identical presentations in students' assignments are evidence of plagiarism and collusion. Such academic misconduct may lead to serious consequences, such as:

- Required to undertake additional assessments in the course
- Failed in the piece of assessment
- Awarded a grade of Fail for the course
- Withdrawn from the course with an academic penalty
- Excluded from the course or the program for a period of time

Refer to USQ Policy \Academic Misconduct" for further details.