

# CPSC 231 Assignment 1

## Eligibility Checker for Alberta Voters

Weight: 3% of final grade

Due date: Friday September 29, at 11:59pm Mountain Time

**Submission:** one Python .py file, submit on the D2L Dropbox. You may submit as many times as you wish, and only the latest submission will be graded.

**Extensions:** You may use your personal days to extend the deadline. An extension request must be submitted using the request form: <https://forms.office.com/r/2wN7KNhYEK>

You have a total of 5 personal days for the entire semester. No penalties for using personal days. An assignment will not be accepted if it is late and all personal days are already used, other than in exceptional circumstances.

**Academic Integrity:** This work must be completed individually. Please follow academic integrity rules as discussed during lecture. You may discuss your ideas in English (not in Python) with other students as much as you like, but make sure that when you write your code that it is your own. A good rule of thumb is to never share your code with anyone, except your instructor and TAs.

### Detailed Descriptions

As a resident of Alberta, you can be granted many rights and privileges. However, these rights and privileges often have conditions attached, and a person must meet the stated conditions. In this assignment, you will write a program that checks the eligibility to vote in Alberta elections.

Step 1:

Create a new Python file to work on.

Step 2:

Your file should start with

```
from datetime import datetime
month = datetime.now().month
day = datetime.now().day
year = datetime.now().year
```

These three variables, *month*, *day*, and *year*, give you the current month, day, and year that you can use later.

You are only allowed to import `datetime` as shown above. You are asked to demonstrate programming at the basic level, so you are not allowed to import anything else.

Step 3:

Create the following behaviour for your program:

Your program should start by asking:

"Are you a Canadian citizen?"

The user can enter "yes" or "no".

"Are you a resident of Alberta?"

The user can enter "yes" or "no".

"What is the month of your birth date?"

The user can enter a month as a number, such as 12.

"What is the day of your birth date?"

The user can enter a day as a number, such as 31.

"What is the year of your birth date?"

The user can enter a year, such as 1990.

Step 4:

After the user enters all the information above, the program should then check for invalid responses, following the order listed below. Do not check for invalid responses until after the user enters all the information.

For the citizen question, if the user's response is anything other than "yes" or "no", output "Invalid response to citizenship." and exit right away.

For the resident question, if the user enters anything other than "yes" or "no", output "Invalid response to residency." and exit right away.

For the month question, if the user enters anything other than an integer between 1 and 12, output "Invalid month." and exit right away.

For the day question, if the user enters anything other than an integer between 1 and 31, output "Invalid day." and exit right away.

For the year question, if the user enters anything other than an integer between 1900 and 2023, output "Invalid year." and exit right away.

#### Step 5:

After the program passes the checks above, check to see the birth date is indeed valid. For example, November 31<sup>st</sup> would be an invalid day since November only has 30 days. Also, a birth date in the future is not valid.

The one tricky part is February 29<sup>th</sup>. You'll need to check to see if the year is a leap year.

If the birth date is not valid, output "Invalid birth date." and exit right away.

#### Step 6:

If the user is less than 18 years old, they are not eligible to vote.

If the user answered "no" to Canadian citizen or "no" to resident of Alberta, they are not eligible to vote.

If the user is 18 years old or older, and is a Canadian citizen and an Alberta resident, they are eligible to vote.

Output either "You are eligible to vote." or "You are not eligible to vote." and exit.

## Test cases

How do you know your program is correct? Your program will be tested on some test cases using an AI auto-grader. The result will determine your grade on the assignment. To help you fix your problems, the auto-grader is provided to you, so you can test your code yourself before you submit. Be sure to do this to make sure your code runs correctly and there are no typos.

To test your code on the test cases:

Step 1: Your Python file needs to be on a lab machine. The auto-grader only runs on a Linux machine.

If your Python file is on your own computer, you will need to copy your Python file to a Linux machine in the CPSC labs.

- If you are sitting in front of a lab machine, you could use a USB stick/flash drive to copy the file over.
- If you are at home, you will need to remotely transfer your file to a CPSC server machine. You could use scp. ([SCP instructions](#))

Step 2: If you are sitting in front of a lab machine, you can open a terminal. If you are not, you will need to use ssh to connect to a CPSC server machine. ([SSH instructions](#))

Step 3: In your Terminal or Bash window, run the auto-grader by typing

```
/home/profs/richard.zhao1/231/a1/autograde {your_file}.py
```

where {your\_file} is the name of your Python file, assuming the file is in your current directory.

You should see the results of the auto-grader.

```
Total correct: 12
Total errors: 0
** Total failed test cases: 0
```

This is to help you find errors in your code only. No one else will see these results. If you are not happy with the results, try to fix your code.

Once you are happy with the results, submit your code on D2L. The code you submit on D2L is the one your TA will grade.

## Grading

Grade Point	Letter Grade	Guidelines
4	A+	Fulfill all assignment specs
4	A	One failed test
3.7	A-	Two failed tests
3.3	B+	Three failed tests
3	B	Four failed tests
2.7	B-	Five failed tests
2.3	C+	Six failed tests
2	C	Seven failed tests
1.7	C-	Eight failed tests
1.3	D+	Nine failed tests
1	D	More than nine failed tests, or code does not run due to syntax errors
0	F	Barely started code, or no submission, or late submission with no extension approved

Your submitted code should be clearly documented with comments. Comments need to include: your name, descriptions of what your code does, and citing any sources you have used to complete this assignment. Code without proper comments could receive a letter grade reduction.

The assignment will be graded out of 4, with the grade based on the program's level of functionality and conformance to the specifications.