

## CPSC 231 Assignment 2

### Examining Trends of Birth Rates Data

Weight: 4% of final grade

Due date: Friday October 20, 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. All personal day extensions are automatically approved after a request form is submitted. An assignment will not be accepted if it is late with no approved extensions, 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.

#### Notes:

- For the purpose of the assignment, you are not allowed to use “break”, “continue”, or “exit()” in your code. Instead of stopping your code halfway when encountering some undesirable conditions, learn to adjust your conditions so that your code runs only in the desirable conditions.
- You are not allowed to use “import” in this assignment so you can practice writing code yourself instead of importing someone else’s code.
- Your code should be written to work with any input from the user, not just the specific inputs tested by the autograder. Basically, you shouldn’t “hard-code” by having in your code something like

```
if input == 1900:  
    print("Bad year.")
```

even if you know the autograder is going to have a test case with 1900 as the input, because this is bad programming practice and your code would be useless if the autograder uses a different set of numbers.

You would be assigned a grade of no higher than C if these rules are not followed.

## Detailed Descriptions

The birth rate is one of the most basic and important measures in demography. Low birth rates can have major impacts on the future of a society – without enough people being born, you either have to accept a large number of immigrants to keep the economy operating, or you have to replace the work that humans did with robots or AI (both are happening in Canada).

This assignment works with some data on birth rates.

Step 1:

Create a new Python file to work on.

Step 2:

**Data entry:** Your code should start by asking the user how many data points they want to enter.

The code then asks the user to enter each data point, based on the number the user wanted. Assume Year is entered as an integer and Birth Rate is a float. Assume birth rate is at most 2 decimals.

As an example:

Year	Birth Rate in Canada
1960	3.81
1970	2.26
1980	1.74

See screenshot:

```
How many data points do you have? 3
What is the year of data point 1? 1960
What is the birth rate of data point 1? 3.81
What is the year of data point 2? 1970
What is the birth rate of data point 2? 2.26
What is the year of data point 3? 1980
What is the birth rate of data point 3? 1.74
```

### Step 3:

Let's add some error checking in the following order.

- When asking the user how many data points they want to enter, assume the user enters an integer. However, code should check to make sure this number is greater than 0. If not, no other code should run, and your code should output "Must enter at least one data point." and stop.
- When asking the user to enter a year, assume the year is entered as an integer. However, code should check to make sure the year is greater than 0. If not, no other code should run, and your code should output "Invalid year." and stop.
- When asking the user to enter a year as part of the data point entry, code should check to make sure the years have been entered in chronological order. You can't enter 1980 before 1970, for example. If not, no other code should run, and your code should output "Years must be entered in order." and stop. *You can achieve this by comparing the currently entered year with the previously entered year.*
- When asking the user to enter a year as part of the data point entry, code should check to make sure the year has not been entered before. You cannot have the same year entered twice. If not, no other code should run, and your code should output "Same year entered twice." and stop.
- When asking the user to enter a birth rate, assume it is a float. However, code should check to make sure the number is 0.0 or greater. If not, no other code should run, and your code should output "Invalid birth rate." and stop.

### Step 4:

**Data analysis:** Now that we have the data, we want to do some analysis on the data. Let's calculate the average birth rate of two given years and calculate the trend.

The next step in the code is to ask the user to enter the two years they want to use. Assume the entered years are integers.

```
Which year would you like to start with? 1960
Which year would you like to end with? 1980
```

Add error checking in the following order:

- If start year does not exist as an existing data point (as entered by user in step 2), no other code should run. The code should output the following message and stop: "The start year does not exist." Check this as soon as the user enters the start year.
- If end year does not exist as an existing data point (as entered by user in step 2), no other code should run. The code should output the following message and stop: "The end year does not exist."
- The end year should be after the start year. If not, no other code should run, and your code should output "End year must be after start year." and stop.

Step 5:

Based on the user entered years, the code will calculate and output the average birth rate of the two years (using only the data from these two years, not any years in between), shown to 2 decimal places. Round if longer (2.337 becomes 2.34); pad 0s if shorter (2 becomes 2.00).

It will also output either “There is an upward trend.” , “There is a downward trend.” , or “There is a sideways trend.” depending on whether the birth rate increased, decreased, or remained the same.

```
The average birth rate of these two years is 2.77.  
There is a downward trend.
```

## Test cases

Your code 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/a2/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.

Once you are ready to submit, submit your code on D2L. The code you submit on D2L is the one your TA will grade.

### Grading

Grade Point	Letter Grade	Guidelines subject to notes stated above
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 code's level of functionality and conformance to the specifications.