COMP202 Assignment 1
Due: October 2, 2019 at 23:59
This is an individual assignment.

A Personal Carbon Footprint Calculator

The main component of this assignment is to create a basic personal carbon footprint calculator. This will allow you to practice writing functions, doing math in Python, working with variables, and calling helper functions.

This calculator shouldn't be taken as an authoritative footprint calculator: we leave a lot of parts of your life uncalculated (e.g. buying furniture, clothes), and many of the calculations are best guesses when local data wasn't available (e.g. for calculating the footprint of using the Metro system here in Montreal we use numbers for the London Underground instead).

The more important thing here is to give you a sense of how programming can be put to good use in the fight against climate change! Also along the way you'll get a sense for the orders of magnitude different activities/items have when it comes to carbon footprint.

Instructions

It is very important that you follow the directions as closely as possible. The directions, while perhaps tedious, are designed to make it as easy as possible for the TAs to mark the assignments by letting them run your assignment, in some cases through automated tests. While these tests will never be used to determine your entire grade, they speed up the process significantly, which allows the TAs to provide better feedback and not waste time on administrative details. Plus, if the TA is in a good mood while he or she is grading, then that increases the chance of them giving out partial marks. :)

Up to 30% can be removed for bad indentation of your code as well as omitting comments, or poor coding structure.

To get full marks, you must:

- Follow all directions below
  - In particular, make sure that all function and variable names are **spelled and capitalized exactly** as described in this document. Else a 50% penalty will be applied.
- Make sure that your code runs.
  - Code that does not run (e.g due to a NameError, TypeError) will receive a very low mark.
- Write your name and student ID as a comment in all .py files you hand in
- Name your variables and helper functions appropriately
  - The purpose of each variable should be obvious from the name
- Comment your work
  - A comment every line is not needed, but there should be enough comments to fully understand your program

What To Submit

Please put all your files in a folder called Assignment1. Zip the folder (DO NOT RAR it) and submit it in MyCourses. If you do not know how to zip files, please ask any search engine or friends. Google will be your best friend with this, and a lot of different little problems as well.

Inside your zipped folder, there must be the following files. **Do not submit any other files.** Any deviation from these requirements may lead to lost marks.

1. `unit_conversion.py`
2. `footprint_services.py`
3. `footprint_transport.py`
4. `footprint_consumption.py`
5. `footprint_calculator.py`
6. $< YourName >$`.csv`
7. `README.txt` In this file, you can tell the TA about any issues you ran into doing this assignment. If you point out an error that you know occurs in your program, it may lead the TA to give you more partial credit.

   This file is also where you should make note of anybody you talked to about the assignment. Remember this is an individual assignment, but you can talk to other students using the **Gilligan's Island Rule**: you can't take any notes/writing/code out of the discussion, and afterwards you must do something inane like watch television for at least 30 minutes.

   If you didn't talk to anybody nor have anything you want to tell the TA, just say "nothing to report" in the file.

1 Warmup: conversion functions [0 points]

In this assignment we will be converting between numerous units, such as lbs to kg, and days to years. This part is not for credit but **these functions will be necessary for the rest of the assignment**.
Download and open the file `unit_conversion.py`. Complete the six functions below. You do need to edit the code outside the functions.

1. **kg_to_tonnes**: convert a mass in kg into one in metric tonnes. 1000 kg is 1 tonne.
2. **pound_to_kg**: convert a mass in pounds into kilograms. 1 lbs is 0.45359237 kg.
3. **km_to_miles**: convert a distance in kilometres to miles. 1 km is 0.621371 miles.
4. **daily_to_annual**: convert a daily quantity into an annual quantity, using Gregorian years. 1 Gregorian year has 365.2425 days.
5. **weekly_to_annual**: convert a weekly quantity into an annual quantity. Again use Gregorian years. One week has seven days.
6. **annual_to_daily**: convert an annual quantity into a daily quantity. Again use Gregorian years.

2 Utilities and Institutions [20 points]

The calculator takes six types of input: utilities, institutions, diet, travel, transportation, and computing. We'll be grouping these into three groups of two to help split up the assignment.

To start off, we'll be calculating the carbon footprints of Québec resident's utilities (hydro/electricity and natural gas) and studying at a university.

Download and open the file `footprint_services.py`. Complete the four functions below. You do need to edit the import statements. For full credit you should not ever be repeating code, but rather calling helper functions!

Important: you may note that in the unit tests we give you, we always round to four decimal places. This is because depending on the order in which you do your operations, you could get results that are very slightly different. Rounding at four decimal places should cut off any subtle floating point differences.

1. **fp_from_gas** [5 points]: based on a monthly natural gas bill (in $), calculates metric tonnes of CO2E produced annually. For every dollar somebody spends monthly, 105 lbs of CO2E is emmitted annually.

   "CO2E" means "CO_2-equivalent", because multiple greenhouse gases (GHGs) contribute to climate change. Methane, for example, is a greenhouse gas that is four times more potent than CO2 (and a large by-product of natural gas and animal agriculture).

2. **fp_from_hydro** [5 points]: based on a daily electricity usage in kWh, calculate the metric tonnes of CO2E produced annually by Hydro Québec. Hydro Québec reports that 0.6 kg CO2E are emitted for each MWh of electriicty. 1 MWh = 1000 kWh.

   You can find your average daily electricity usage by logging in to the Hydro Québec website. In Québec the footprint of our electricity usage is very low relative to global standards! This is because we use mostly (about 96%) hydroelectricity, whereas in other parts of the world the standard is to make electricity from coal (which is very carbon intensive).

3. **fp_of_utilities** [5 points]: based on a monthly natural gas bill (in $) and an average daily electricity usage in kWh, calculate the metric tonnes of CO2E produced annually.

4. **fp_of_studies** [5 points]: based on an annual number of credits, calculate the metric tonnes of CO2E from being a student at McGill. McGill emits annually 56,004 tonnes CO2E — most of it from powering and maintaining its buildings.

   Not all students use the campus the same amount. In university budgeting you'll commonly see the unit "FTE" for full-time equivalent. One full time student (30 credits) is 1 FTE; a student taking 15 credits would be 0.5 FTE.

   Per McGill's 2017 Greenhouse Gas Inventory, the per-student emissions work out to 1.12 annual tonnes per FTE.

Naming convention: the footprint of everything from a category (e.g. utilities, travel) will be "fp_of_". Helper functions which calculate the footprint of things in that category will be "fp_from_".

3 Travel and Transportation [30 points]

Download and open the file `footprint_transport.py`. Complete these five functions. You do need to edit the code outside the functions. For full credit you should not ever be repeating code, but rather calling (and creating) helper functions!

1. **fp_from_driving** [5 points]: based on annual number of kilometers driven, calculate in metric tonnes of CO2E this produces.

   To get the total *pounds* of CO2E somebody gets from driving a given number of *miles*, multiply the mileage by 0.79.

2. **fp_from_taxi_uber** [5 points]: based on weekly average number of taxi/Uber/Lyft/etc rides, calculate annual metric tonnes of CO2E produced.

   For this we'll note that 81 million Uber rides produces 100,000 metric tonnes of CO2E, and assume that taxis/Lyft/etc have the same footprint.

3. **fp_from_transit** [5 points]: based on weekly average number of bus and rail (metro/exo) one-way rides, calculate annual metric tonnes of CO2E produced. You'll want to know that:

   (a) The average transit trip in Montréal is 7.7 km. We'll assume all trips have this length.

   (b) One mile by bus: 150 g CO2E

   (c) One mile by subway train: 160 g CO2E

4. **fp_of_transportaton** [5 points]: given somebody's weekly bus/rail/Uber trips, and their weekly km driven calculate the annual metric tonnes of CO2E produced.

5. **fp_of_travel** [10 points]: given how many annual one-way flights, intercity train rides, intercity coach bus rides, and spending (in $) on hotels one does, calculate the annual metric tonnes of CO2E produced.

   You'll want to create helper functions to break this up! Here are things you'll want to know:

   (a) One short (4 hours or less) flight: 1,100 lbs CO2E

   (b) One long (4 hours or more) flight: 4,400 lbs CO2E

   (c) One train ride with Via: 34.45 kg CO2E

   (d) One coach bus ride: 33 kg CO2E

   (e) For each dollar spent on a hotel stay: 270 g CO2E

4 Computing and Diet [20 points]

Download and open the file `footprint_consumption.py`. Complete these two functions. You do need to edit the import statements. For full credit you should not ever be repeating code, but rather calling (and creating) helper functions!

1. **fp_of_computing [10 points]**: given how many hours a day on average you spend online, how many hours a day you use your phone, and how many new devices you bought this year, estimate the annual metric tonnes of CO2E this produces.

   For computing, most of the footprint comes from the manufacturing of your devices rather than their everyday usage. The materials used to make a laptop/phone/etc are carbon-intensive to mine and then assemble! And then for our phones, there's an additional footprint from using the telecommunications infrastructure needed for connectivity (e.g. phone towers, satellites).

   You'll want to create helper functions to break this up! Here are things you'll want to know:

   (a) One hour of being online: 55 g CO2E

   (b) One year of using your phone for one hour a day: 1250 kg CO2E

   (c) One new small portable device (e.g. phone, e-reader, small tablet): 75 kg CO2E

   (d) One new medium-sized device (e.g. laptop, big tablet): 200 kg CO2E

   (e) One new large/heavy device (e.g. desktop workstation, server, or gaming console): 800 kg CO2E

2. **fp_of_diet [10 points]**: given the average daily consumption of meat (in g), cheese (in g), milk (in L), and eggs, estimate the annual metric tonnes of CO2E this diet produces.

   Again, for full credit you'll want to break this up and use helper functions! Here's what you need to know:

   (a) A vegan (no eggs/dairy/meat) diet produces 2.89 kg CO2E a day (just from the food)

   (b) Adding meat, dairy and eggs to the diet adds:

      i. One gram of meat: 26.8 g CO2E

      ii. One litre of milk: 267.7777 g CO2E

      iii. One gram of cheese: 12 g CO2E

      iv. One egg: 300 g CO2E

   Sidenote: A more refined carbon footprint calculator would also take into account food waste, and how much of the food is air freighted (e.g. asparagus from Peru is carbon-intensive because it is flown in, whereas some bananas from Latin America that were shipped by sea are actually quite carbon-efficient.)

5 Bringing It All Together [5 points]

Download and open `footprint_calculator.py` and `example.csv`. Now we want to bring all our functions from the earlier parts together to calculate one annual personal carbon footprint. Open the file `example.csv`. You'll see it has one imaginary person's input to our calculator, such as their weekly bus trips and their average daily hydro consumption.

TODO: [1 point] Make a copy of `example.csv`, with your first name as the file name followed by `.csv`. Go through the file: change the values to your own estimates of your activities/consumption!

The code provided to you in `footprint_calculator.py` takes in a csv file such as `example.csv` or one formatted like it, and calculates the carbon footprint in metric tonnes of CO2E for each category. You have one function to complete in the file:

1. `input_filename` [4 points]: ask the user to input the name of the person getting their footprint calculated (for the prompt, use the provided variable `PROMPT`), and return the filename (e.g. "Cat" to "Cat.csv")

With this done, run `footprint_calculator`. Compare your footprint to that of `example.csv`. Here's what you should get from `example.csv`:

```
Name of person (file must be in same directory): example
Category Tonnes CO2E (sum: 15.4591)
Utilities        0.9632
University       1.008
Computing        6.2304
Diet             1.4046
Transportation   0.6085
Travel           5.2444
```

5.1 Closing Notes

The book *How Bad Are Bananas: The Carbon Footprint of Everything* was a frequent source for the calculations in this assignment. In the book, the author Mike Berners-Lee argues that North Americans need to reduce our carbon footprint to 10 tonnes CO2E per year. Right now the average Canadian is responsible for 22 tonnes of CO2E a year.

Our footprint calculator is really a "toe-print": to keep the assignment to a reasonable size and difficulty, we're not very comprehensive in tabulating all the possible sources of emissions somebody could have. The result is you'll get a lower bound for your footprint. *It also means that if you get a value above 10 tonnes CO2E, you're already over this 10 tonne budget!*

Finally, it's important to note that mitigating climate change cannot be seen as something that only requires individual behaviour change. Transit, for example, becomes a more viable option for people when cities invest in better public transit. People won't buy so many new laptops if tech companies build laptops to have longer lifespands.

Modelling can help us identify where in society we can make change that will have the most effects. Policy makers and advocates often look to models and projections to decide what to prioritize. If you want to learn more about potential solutions to climate change and estimating their differential effects, I (Elizabeth) recommend reading the book *Drawdown* by Paul Hawken!