

## Programming Project 03

### Assignment Overview

This assignment focuses on the design, implementation and testing of a Python program which uses file processing, functions, lists and tuples to solve the problem described below.

### Assignment Specifications

The craters on Venus are the best source of Space Dust, which is coveted throughout the known universe for its healing qualities. Your employer, the Green Thumb Strip Mining Corporation, has tasked you with compiling a list of the craters of Venus to be mined. Fortunately, you've found an old database developed by NASA in the late 1990s, shortly before the outbreak of the First Zombie War.

Unfortunately, you have to do some work with it since not all craters are created equal, when it comes to the environmentally safe nuclear blasting that your employer will need to extract the Dust. Your employer has a government contract permitting exclusive rights for crater exploitation between latitude -40 and 50 degrees and between longitude 40 degrees and 135 degrees. Furthermore, only craters of at least 60km diameter may be mined. You will process the original input file, named `rel3main.txt`, and write all eligible craters to an output file named `craters.txt`.

For this assignment, you will create four functions:

`def get_crater_tuple(line_str):` This function has a string as a parameter and returns a tuple of the form (ID, name, latitude, longitude, diameter), where ID is an int, name is a string, latitude is a float, longitude is a float, and diameter is a float. The desired parameters are at the beginning of the line—check the file headers to get the order.

`def read_craters(filename):` This function accepts a filename, which is a string, and returns a list of crater tuples. This function will read the file specified by filename, and call the `get_crater_tuple` function on individual lines to obtain tuples to add to the crater list. If the file doesn't open correctly, reprompt for a filename until the file opens correctly.

`def get_eligible_craters(crater_list):` This function takes a crater list and returns a new list, preserving only entries that meet the latitude, longitude, and diameter requirements listed above. Note that this function does not modify the crater list, but instead creates and returns a new list.

`def write_craters(eligible_crater_list):` This function takes an eligible list of craters and writes them to a file named `craters.txt`. Nothing is returned. Remember to close the file.

The main part of your program should simply call these functions as follows:

```
filename = input("Enter a filename: ")
crater_list = read_craters(filename)
eligible_crater_list = get_eligible_craters(crater_list)
write_craters(eligible_crater_list)
```

### Assignment Deliverables

The deliverable for this assignment is the following file:

proj03.py – the source code for your Python program

### Assignment Notes

1. The main part of your program should prompt for a file and call `read_craters` to fill the crater list. It should then call `get_eligible_craters` with that crater list to get a filtered crater list. Finally, it should pass that filtered crater list to the `write_craters` function.
2. As noted above, you are required to define and use four functions, but you may define and use additional functions, if you wish.
3. A small text file named `small.txt` is included for you to experiment with before running your program on the large `rel3main.txt` file. Note that there are header lines that you need to skip and the `file.readline()` method is useful for reading and ignoring a line. You will need to look at the file to figure out which columns contain the data you are interested in—there is a lot more data than you need.
4. The `craters.txt` file should be formatted as follows (this is the output of `small.txt`):  
ID: field width of 3  
Name: field width of 15  
Latitude, Longitude, and Diameter: field width of 9 with 2 digits of precision  
Each field is separated by a space.

ID	Name	Latitude	Longitude	Diameter
1	Mead	12.50	57.00	268.70
9	Joliot-Curie	-1.70	62.40	100.90
14	Potania	31.70	53.00	89.60
15	Bonnevie	-36.10	127.00	87.60
21	Boulanger	-26.60	99.30	73.70

Refer to class notes on formatting of Python outputs. You could also read this tutorial on formatting of python output: <https://realpython.com/python-formatted-output/>