

## Closed Lab 09

*University of Wisconsin – La Crosse**April 8*

Description: In last week's lab, you created a simple class called **Mountain** with several data elements and methods. For this assignment you will add one data element (called `note`) and 10 more methods, most of which are special. As a reminder, the data elements are in the table below.

Attribute	Type
<code>name</code>	string
<code>elevation</code>	integer
<code>prominence</code>	integer
<code>latitude</code>	tuple
<code>longitude</code>	tuple
<code>climbed</code>	boolean
<code>note</code>	string

`climbed` is initially `False` for all instances. In addition, the **Mountain** class has a class variable called `num_mountains`. The class should live in a file called `mountain.py`.

Your class also needs some methods. Below is a list:

- `is_climbed` returns a boolean indicating whether you've climbed the mountain
- `__gt__` based first on elevation then on prominence if elevations are equal

- `__lt__` symmetric case of `__gt__`
- `__bool__` True if the mountain has an elevation of at least 2000m (otherwise it's only a hill)
- `__iadd__` increases the elevation by int parameter `gain`
- `__isub__` decreases the elevation by int parameter `loss`
- `__str__` returns a string for pretty printing a mountain instance
- `__repr__` returns a string that can be used to recreate the instance
- `__call__` takes a string as a parameter and assigns it to `note`

When your class is complete, create a file called `mountain_driver.py`. In this file, write a program that creates a list of `Mountain` instances. The list is called `swiss_mountains`. Your program will read the contents of the file `mountains.txt`. Each line of the file contains the relevant data for one stunningly beautiful hunk of granite. Populate the list with the mountains and test all of the methods in the class. Finally, you should write a loop that prints the data for all of the mountains.