

HW 1

Submit Assignment

Due Tuesday by 11:59pm **Points** 50 **Submitting** a file upload

Edit (1/28): Your canvas turn-in should include one text file with the link to your github repo and your two python files, mostly as a backup to the github mechanism that we are using. You should also commit and push your code to the repo and share the repo with the instructors.

In this assignment you will gain some familiarity with the git revision control system and the Python programming language. These are both core infrastructure that will be necessary to succeed in the rest of the class.

1. If you don't already have one, create a (free) github account, using your UNCC email address.
2. Go to the class repo
 - git clone <https://github.com/jeremy-u/ECGR4090-ML-IoT.git>
3. Fork the "ML-for-IoT" repo.

◦ `git clone https://github.com/jeremy-u/ECGR4090-ML-IoT.git`

4. Invite the instructors (via their github accounts "jeremy-u" and "mhasan13-here") to your repo. Settings->Manage Access->Invite a collaborator.
5. Complete the programming assignments below.
6. Commit and push your assignments.

Programming tasks - list comprehensions, dict, numpy, matplotlib, module

- A. Run the starter script hw1p1.py. It should print out several names and tell how long each name is.
- B. Use a list comprehension to create a list of lengths of the names. Here is an example list comprehension. This is quite a powerful feature; search around the web for more complex examples.

◦ `new_list = [f(x) for x in original_list]`

- C. Create a separate file, named "person.py". In that file, define a class "person" with fields "name", "age", and "height." Define the `__init__` method (constructor) that takes arguments 'name', 'age', and 'height', and sets the object variables of the same names. Eg.

◦

```
new_person = person(name='Joe', age=34, height=184)
print("{:} is {:} years old.".format(new_person.name, new_person.age))
```

should create an object named "new_person" with the name, age, and height indicated and then print out "Joe is 34 years old."

- D. Add a method `__repr__(self)` that returns a string similar to "Bob is 17 years old and 182 cm tall".
- E. Back in the main file hw1p1.py, add a line to "import person". Then iterate through the names and create a dictionary named "people" where the key is a name, and the value is a person object.
- F. Use `numpy.array()` to convert the list of ages into an array. Repeat for the list of heights.
- G. Use the `numpy.mean()` function to find the average age of the people in your list. Print the age using a print function and an f-string.
- H. Use matplotlib to create a scatter plot of ages (on the x axis) and heights (on the y axis). Add gridlines, x and y axis labels, and a title. Export the plot as a png file.
- I. Commit your code changes and push them back up to your github repo.

