1 Lab Week 06 Methods and Lists and Strings

1. Write a function `target_times` that takes the times taken in a race and *prints out* the target times for each competitor for the next race. Each competitor needs to improve their time by 10%.

```
time_taken = [32, 27, 23, 26, 26, 18]
names = ["Ann", "Joe", "Pat", "Ken", "Ali", "Ger"]
target_times(time_taken, names)
```

2. (i) Write a function `get_months_beginning_with` that takes 2 parameters
   - a list of months
   - a letter

   The method creates and returns a list of all the months that start with the required letter.

```
months = [ "January", "February", "March", "April", "May", "June", "July", "August",
"September", "October", "November", "December"]

j_months = get_months_that_start_with(months, 'J')
print(j_months)
```

Hint: to check if a string starts with the letter 'j'

```
    name = "Joe"
    if name.startswith('J'):
        print("yes")
```

(ii)Write a function `get_short_names` that takes 1 parameter which will be the list of months. In the method create a new list and add an abbreviated version of each month to the new list. Return this list.

This is how you will test the function.

```
abbrev_names = get_short_names(months)
print(abbrev_names)
```

Hint: To shorten the name of each month use slicing ie. `word[0:3]`

The program should print out the abbreviated list which looks like this:

```
["Jan", "Feb", "Mar",… ]
```

3. Write a method `get_overfives` that takes in the following two lists and *returns* two new lists with the names and ages of the children over 5.

```
names = ["mary" , "ann",  "ben" , "kate",   "emily" ]
ages = [4, 7, 9, 2, 1]
names_over_five , ages_over_fives = get_over_fives(names, ages)
print(names_over_five)
print(ages_over_fives)
```

2

1. Complete the code and methods for Megan's Barrista.

   Use the hardcoded list given for `hours_worked` while writing all the other methods.

   You can modify `get_hours_worked` at the end to get the data from the user.

2. Write a function `get_three_letter_words` that takes a line of words as a parameter and returns a list of all the three letter words in this line. Get your program to print out the number of three letter words found.

```
line = "i know some new tricks, said the cat in the hat. a lot of good
tricks."
threeLetterWords = get_three_letter_words(line)
```

Hint: The function must convert the line to a list of words using the string method `split` . The function should then create and return a subset of these words in another list.

Example using split:

```
sentence = "cat in hat"
list_of_words = sentence.split(sentence)
```

3. Write a function `get_words_of_specific_length` that takes 2 parameters
   a. a string
   b. required word length

The function returns a list of words that are the specified length found in the string.

For example a call to

```
get_words_of_specific_length("I do not like green eggs and ham", 4)
```

would return the list `[like, eggs]`.