CIS 658 – Getting Comfy with Ruby!
Winter 2019
Homework #2

Work in pairs during class to complete as many of the problems below as possible. You may use any ruby interpreter version 2.0 or more recent.

Due: Jan 24, 2019.

Write the Ruby code necessary to accomplish the following:

1. Print out the ubiquitous “Hello, World” message.

2. For the string “Hello, World,” find and print the index of the word “World”.

3. Write a loop that prints the string “This is funny monkey #1!” 10 times where the number 1 changes from 1 to 10. Implement this in 3 *different* ways using Ruby (including both “normal” and functional styles).

4. Write a simple game that generates a random number 1 – 1000. Let a player guess the number. If the guess is wrong print out whether the guess is low or high and let the player guess again. Repeat this until the user guesses the number. Award the lucky user who finally gets it right, a surprise of their choice. Hint: `rand(1000)` will generate a random number random number 0 .. 999. The function `gets` will read a string from the keyboard that can be translated into an integer.

5. Given the following array definition in Ruby, generate a method named `convert_to_type_strings` that takes the array as input and returns a second array where each element in the array corresponds to a string representation of the type (e.g. class) of each element in the array.

```
data = ['hello', 0, :sym, 3.4, "world", true, [0..3]]
```

e.g, the method should return the following array:

```
["String", "Integer", "Symbol", "Float", "String", "TrueClass", "Array"]
```

6. Augment the existing `Array` class in Ruby so that it has a method named `convert_to_type_strings` that does exactly what the method in question #5 above does but using the array's internal data (e.g. you cannot pass the array as a parameter). Hint: investigate the `map` method defined by `Array`. When you are finished, test with the following code:

```
data = ['hello', 0, :sym, 3.4, "world", true, [0..3]]
data.convert_to_type_strings
```

e.g. the call to convert_to_type_strings on the array should produce:

```
["String", "Integer", "Symbol", "Float", "String", "TrueClass", "Array"]
```

7. Given an array of symbol values representing an ensemble, write a method called `tabulate_sections` that produces a hash that maps a string representation of the section (e.g. type) the instrument belongs to, to the number of instruments in that section of the ensemble. For example, the input array:

```
ensemble = [:piano, :clarinet, :oboe, :trumpet, :frenchhorn, :violin, :piano,
     :oboe, :cello]
```

produces the output hash:

```
{"percussion"=>2, "woodwind"=>3, "brass"=>2, "strings"=>2}
```

You may assume that you only need to deal with the instrument and instrument types referenced in this example.

Submission instructions:

Option 1 - Combine your individual solutions of Ruby into a single text file with a simple comment indicating which homework problem each fragment of Ruby solves. Have one team member submit the file via Slack in a direct message to the instructor and the remaining team member.

Option 2 – Push your Ruby code to a github repo and have one team member post the URL of the repo as a direct message to the instructor and the remaining team member on Slack.