CS/INFO 3300 // INFO 5100
Homework 1
Due at 11:59pm Wednesday, August 31

Goals: Review Javascript types and program flow.

Your work should be in the form of an **HTML file called `index.html` with one <p> element per problem**. Wrap any Javascript code for each problem in a `<script>` element nested within the <p> element.

For example:

```
<html><body>
  <p id="p0">
    Problem 0: We use a "let" statement to...
    <script>
      let x = 100;
    </script>
  </p>
</body></html>
```

Create a zip archive containing this file and upload it to CMS before the deadline.

1. For each of the following items in the list, please:
   A: Identify whether it is a **canonical Javascript type**
   B: If the type is a canonical Javascript type, please **provide an example** in your `<script>` section using `console.log(typeof( ... ))`.
   C: If applicable, identify any unusual or unexpected `typeof()` behavior for the type. (do this whether or not it is a canonical JS type if it is applicable)

   a) dingus; b) array; c) boolean; d) int; e) number; f) class;
   g) regex; h) string; i) Unicode; j) object; k) null; l) function;
   m) NaN (not a number); n) char; o) pointer; p) undefined;

2. First, in 1-2 sentences, please **define type coercion**. Then, in your <p> tag please provide two examples where type coercion **produces an answer that is unusual or unexpected** (i.e. different from Python output or a case where Python would throw an exception). For each sample, briefly explain why it is unexpected. Then, in the `<script>` section put both of your code examples into of a `console.log()` statement so we can see the result (e.g. `console.log("apple" + "orange")`)

(next page)

3. In your `<script>` tag, **create a <u>recursive</u> function**, `fact( n )`, **which takes an integer**, `n`, **as an argument and returns the factorial of** `n`. Do not use loops or the built-in factorial functions to accomplish this. Remember that recursion involves two major components: base case(s) and recursive calls of the same function.

Make sure that your `fact` function **handles negative integers by returning** `undefined`, properly **returns 1 for** `fact( 0 )`, and otherwise **uses recursion to compute correct factorials** for larger integers.

Include the following code block after your function to prove it works as expected:

```
        console.log( fact(-2) );
        console.log( fact(0) );
        console.log( fact(1) );
        console.log( fact(5) );
        console.log( fact(12) );
```

(As the function is only intended to handle integers, do not worry about implementing the gamma function for decimal numbers)

4. Describe, in your own words, the course policy on late and unreadable work outlined in the syllabus, which is linked from the course website.