CS2613: Programming Languages Laboratory
Final Exam – Practice Exam

Student Name: _____

Student Number:_____

Rules:

- Some questions will require you to complete on paper, others will require you to submit via **email as a .zip**. View table below for full list of Submission Methods.
    - If you wish to complete Q3 or Q4B on paper, please request an exam booklet.
    - You should make one submission via email when you are finished.
- All work is to be completed individually.
- You may not use the Internet for this Exam.
    - D2L or Cloud File Storage allowed to access programs/notes you created.
    - Required files can be found on D2L.
    - Submissions for online questions are via D2L dropbox.
- If you have a question, you must raise your hand.
- If you require to go to the washroom, you must hand over any digital devices to the instructor.
    - This includes, but is not limited to, phones, tablets, computers, and smartwatches.
- Once complete, please quietly leave the lab room.

Breakdown:

#	Submission Method	Out of	Scored
1	Paper	8	
2	Email	14	
3	Paper or Email	8	
4A	Email	12	
4B	Paper or Email	8	

Total Grade: _____/50 = _____%

Good Luck!

1. **[8 points total, 2 point each]** For each excerpt of code below, you are provided with a description of what the code should do as well as the returned/output value. Each question involves at least two blanks, what must go in the blanks to attain the provided returned/outputted value?

   a. **Python**: The following code should only print strings that start with a capital "K" **<u>or</u>** have four characters.

```
rem = (lambda n: True if (_____________________________________) else False)

shortened = _____________(rem, ["Anne", "Jeff", "Amy", "Kyle"])

for val in shortened
     print(val)
```

   b. **Racket**: The following function "`apply-compare`" should apply a function to each value in a list and return a list. The length of the return list should equal to the number of values that are the same in both lists at each position.

```
#lang racket
(define (comp x) (= x 0))

(define (subtract x y)
  (cond [(<= (length y) 0) '()]
  [else (cons (- (list-ref x 0) (list-ref y 0)) (subtract (drop x 1)
(drop y 1)))]))
  )

(define (apply-compare l1 l2 func)

  (filter _______________

      (subtract (_____________________) (_____________________)))
  )
```

Example Function Call:

```
(length (apply-compare '(1 2 3 4) '(1 2 2 4) add1)) = 3
```

c. **JavaScript**: Write a function using the arrow operator that takes 2 arrays of equal length and 1 function as a parameter that returns a value. This function returns an array of the same length where each element is the higher of the two values in the parameter arrays.

```
const filtered = ____________________________________________{
      let a3 = [];
      for(let i = 0; i < a1.length; i++){

            a3[a3.length] = ____________________________________;
      }
      return a3;
}

const higher = ____________________________________{
      if(v1 > v2){
            return v1;
      }
      else{
            return v2;
      }
}
```

Example Function Call:

```
filtered([3, 6, 1], [6, 2, 5], higher) = [ 6, 6, 5 ]
```

d. **Octave**: The following code should print a list of the summation of all values in the list raised to the power of 2.

```
f1 = _____________________________;

function retval = summ(func, lst)
      retval = 0;
      for i = 1:length(lst)

            retval = ____________________________________________;
      end
end

list = [1,2,3,4];
summ(f1, list)
```

2. **[14 points total, 7 point each]** For each question below, create a program and submit your file **via D2L as a .zip** named "Exam_Q2X" where X is replaced with A or B. Each question has a **required** programming language.

   a. Write the following 3 functions in Python. All functions **must use lambda notation**. Include at least one test for each function.
      1. A function that takes a numeric value greater than or equal to 0. You do not need to check that the value is in the proper format. The function should return the final digit in of the passed in numeric value. Examples: 155 = 5, 632 = 2, 0 = 0.
      2. A function that takes a value and returns true if the value is within the range of [0, 100] (inclusive).
      3. A function that takes *at least one* parameter being a list of values and **recursively** (directly or indirectly) returns a string that contains all values of the list with tab characters between each value.

   Now create a list of numeric values in your program. Use function 1 to map the list and then filter the list using function 2. You must use Python's `map` and `filter` functionalities to receive full points on this question. Finally, print the list using the third function you created.

   Example Recursive Lambda Function:

```
func_name = (lambda x: func_name(x))
```

   b. Write a recursive (direct or indirect) function in **Racket** named "piecewise" that takes no parameters and returns a list of all values [3, 25] after being calculated in the piecewise function found below. A functional programming approach is ***not*** required; however, you may find that writing some additional functions is useful.

$$f(x) = \begin{cases} x!, & x < 7 \\ \frac{x * 5}{2} + 1, & 7 \leq x < 12 \\ -x, & 12 \leq x < 20 \\ x * 100, & \text{else} \end{cases}$$

   Example Function Call:

```
piecewise() = '(6 24 120 720 18½ 21 23½ 26 28½ -12 -13 -14 -15
-16 -17 -18 -19 2000 2100 2200 2300 2400 2500)
```

3. **[8 points]** On D2L, you have been provided with an Octave code file entitled "Question3.m" as well as two .csv files for testing purposes. Download these files and open them. The code contained in this file does not function properly due to various syntax errors (errors that will not allow the program to run) and semantic errors (logical errors that cause the output to be incorrect). It is your job to identify the errors and give feedback to help the programmer ***learn*** from their mistakes. You should assume that giving the programmer the correct answer only will not help them learn. For full points, identify **at least four** unique issues the programmer encountered and **fully explain** why the problem occurred. You are ***NOT*** required to correct the original code, but you may find it useful.

   You may complete this problem digitally (submit a PDF via email) or on paper (request an Exam Booklet).

   Problem: This code is an attempted solution to Octave Programming Question 1.

4. **[20 points total, 12 points for part A, 8 points for part B]**
   a. Write a program in any of the four languages (Python, JavaScript, Racket, or Octave) that meets the description below. You must select the language with consideration as you will be asked to describe the reason you made your choice in Part B. Simply saying "I chose X because I am most familiar with it" will not yield a high grade.

      Create an Object-Oriented Binary Search Tree (BST) program. This BST should contain Song objects. Each Song should have three data points – the Song's name, Song's artist, and the length of the Song in seconds. For this program, songs are ordered by their running time. The only *required* method for a Song object is a "toString()" method or equivalent; however, you may wish to add other methods.

      Your Binary Search Tree object should have 3 methods plus a constructor (minimum):
      1. **Insert** – takes a Song as a parameter and adds it in order of the Song's length.
      2. **Search** – searches the tree for a Song and returns True if the object is in the tree or false otherwise.
      3. **Print** – prints the nodes of the Binary Search Tree **in order**.

   b. Why did you select the language that you did? Use your knowledge of the four languages and their intricacies to guide your answer rather than solely your opinion. An answer that receives full points should thoroughly convince the reader that you made the correct choice. Describe the reason for your selection over the other three languages.

      You may complete this problem digitally (submit a PDF to the Dropbox "Exam_Q4B") or on paper (request an Exam Booklet).

--------- Have a good winter break! ---------