# *Advanced Topics in Computer Science II (420-G50-HR)*

# *Lab 1 – JavaScript and Python Review*

Date due: **January 24, 2025**

**Learning Objectives**

Upon successful completion of this lab exercise, the student will be able to:

* Review JavaScript
* Review Python

Lab Set-Up

1. Create a folder called *initials*G50L01 where *initials* are your initials. Unzip the contents of the zip file you downloaded to this folder.
2. I advise you to use VS Code as the editor. However, feel free to use any other IDE of choice if you prefer.
3. It is important that you name the files and functions as requested. The attached test programs are expecting certain file names, or they will not work.
4. All Python code MUST be Pythonic and follow PEP-8 standards. All JavaScript code must also adhere to proper ES6 coding standards.
5. For both languages use functions that do ONE thing and do not repeat code.

To Do

**Part A**

You are going to complete Part A in both **Python** and **JavaScript**. The PartA folder has a Python folder and a JavaScript folder. Each contains the files you need to test your code. Create a program called parta in the appropriate folder. Use the appropriate extension for the type of file (use py for Python and js for JavaScript). There are test files for both languages. For JavaScript no need for a separate folder since the HTML code really does nothing except read in the JavaScript files.

1. Create a function **final\_grade** (**finalGrade** in JavaScript), which calculates the final grade of a student depending on two parameters: a grade for the exam and a number of completed projects.
   1. This function should take two arguments: exam - grade for exam (from 0 to 100); projects - number of completed projects (from 0 and above). You don’t need to check for errors.
   2. This function should return a string (final grade). There are four types of final grades:
      1. A, if a grade for the exam is more than 90 or if a number of completed projects more than 10.
      2. B, if a grade for the exam is more than 75 and if a number of completed projects is minimum 5.
      3. C, if a grade for the exam is more than 50 and if a number of completed projects is minimum 2.
      4. F, in all other cases.
2. Create a function called **phone\_number** (**phoneNumber** in JavaScript) that accepts an array of 10 integers (between 0 and 9), that returns a string of those numbers in the form of a phone number.
   1. Return the string “error” if there is an error.
      1. Error if not exactly 10 elements in the array
      2. Error if array elements not single digit numbers
   2. Returned non-error string should be formatted as (###) ###-#### where the # are the digits in the array from 0 – 9. If you want to use reduce for this, you need to import **functools** in Python.
3. Create a function called **readable\_time** (**readableTime** in JavaScript) that takes a non-negative integer (seconds) as input and returns the time in a human-readable format (HH:MM:SS).
   1. No error checking is required
   2. HH is hours, MM is minutes and SS is seconds
   3. If value of minutes or seconds is only one-digit (0-9) add a 0 to the front
   4. No error checking is required
4. Add a function called **people\_on\_bus** (**peopleOnBus** in JavaScript) that counts the number of people on a bus at the end of the bus’ route.
   1. There is a bus moving in the city, and it takes and drop some people in each bus stop.
   2. You are provided with a list (or array) of integer pairs. Elements of each pair represent number of people getting into the bus (the first item) and number of people get off the bus (the second item) in a bus stop.
   3. Your task is to return number of people who are still in the bus after the last bus station (after the last array element). Even though it is the last bus stop, the bus is not empty and some people are still in the bus, and they are probably sleeping there :D
   4. Please keep in mind that the test cases ensure that the number of people in the bus is always >= 0. So, the return integer can't be negative.
   5. The second value in the first integer array is 0, since the bus is empty in the first bus stop.
   6. No error checking is required.

**Part B – Wordle**

1. Create a Python program called wordle.py to play the Internet’s latest craze: Wordle. This part is a good review of file I/O and formatting in the console.
2. Wordle explained:
   1. Randomly select a 5-letter word with no repeated letters (a list of words is provided)
   2. Let the user enter a guess
   3. Redisplay the user’s guess as follows:
      1. If the letter is correct and in the correct spot, display the letter in green
      2. If the letter is correct and in the wrong spot, display the letter in blue
      3. If the letter is not in the word, display it in the default colour
   4. For example:
      1. WEARY – the W is in the word and in the correct spot
      2. PILOT – the L is in the word but in the wrong spot
      3. VAGUE – None of the letters are in the word
3. What you need to do:
   1. Create a file called wordle.py
   2. Read in the file FiveLetterWords.txt
   3. Select one of the words at random
   4. Prompt the user for input
   5. Upper case the input and compare it letter by letter to the random word
   6. Using **colorama** display the letter in green, blue or default depending on the above
   7. If all letters match, congratulate the user and exit
   8. If all letters do not match, loop to user input

**To submit**

When you have completed the lab exercise, call the Teacher’s attention and we’ll go over it together. Then, create a single zip file called *initials*G50L01.zip and submit the file to the Moodle page for the course.