CS 161 Lab #7 – 1-D Arrays and C-style Strings

- Get checked off for **(up to) 8 points** of work from the previous lab in the first 10 minutes, **if you had a non-zero grade for Lab 6 already**.
- **To get credit for this lab, you must be checked off by a TA by the end of lab.**
- **This lab should be done solo, not via pair programming.**

Goals:

- Practice creating, accessing, and updating 1-D arrays
- Practice working with C-style strings
- Practice passing arrays to functions

(3 pts) A. Lab Quiz (Canvas)

Visit this link on Canvas to take the Lab 7 quiz:
https://oregonstate.instructure.com/courses/1771939/quizzes/2535322

Re-take the quiz until you get all of the questions right! Canvas saves your last score, not your highest score. If you don't get 100% within the time available, finish outside of lab.

(2 pts) B. Working with 1-dimensional Arrays

1. Static arrays: Tracking fruit fly populations

Design a program to allow a scientist to enter the number of fruit flies in each of 5 jars. After the data has been entered, the program should print out the index of the jar with the largest population and the index of the jar with the smallest population.

Before writing the program, create 3 test cases (with expected output). Put your test cases in comments in the program.

After writing the program **(lab7_fruitfly.cpp)**, run your test cases and check that they work correctly. You do not need to check user input for validity (you will be the user ☺).

Example output (user input is highlighted):

```
Hello Scientists! You have 5 jars to fill with fruit flies.
How many fruit flies would you like in jar 0: 3
How many fruit flies would you like in jar 1: 2
How many fruit flies would you like in jar 2: 5
How many fruit flies would you like in jar 3: 1
How many fruit flies would you like in jar 4: 4

The jar with the most fruit flies: jar 2
The jar with the least fruit flies: jar 3
```

2. Dynamic arrays: Tracking rainfall for an unknown number of cities

Design a program to allow a meteorologist to enter the number of inches of rainfall received in the last week for several cities. The program should ask the meteorologist how many cities they want to enter data for, allocate an appropriately sized array, and allow them to enter the data. After the data has been entered, the program should print out the average number of rainfall inches across all of the cities. **Be sure to clean up the heap before the program ends.**

Before writing the program, create 3 test cases (with expected output). Put your test cases in comments in the program.

After writing the program **(lab7_rainfall.cpp)**, run your test cases and check that they work correctly. You do not need to check user input for validity.

Example output (user input is highlighted):

```
Hello Scientists! You need to record the rainfall per city (in inches).
How many cities would you like to record the rainfall of: 3
How much rainfall was there in city 0: 6.2
How much rainfall was there in city 1: 1.3
How much rainfall was there in city 2: 2.7

Average rainfall per city: 3.4 inches
```

Is it possible to write this program without using dynamic memory (heap)?

(3 pts) C. Working with C-style Strings

For the rest of this lab, each time "string" is used, it means "C-style string".

1. Design a program to read in a string (max 20 characters) from the user and report the percentage of characters in the string that are the character '*' (let's call this the "star-factor"). The program should print out the result.

Example output (user input is highlighted):

```
Please enter a string (<20 chars): ***Hydra*** *Virgo*
String length: 19
Stars: 8
Star-factor (percentage): 42.1052%
```

2. What are useful test cases? Before writing the program, create 3 test cases (with expected output). Put your test cases in comments in the program.

3. Implement your program **(lab7_star.cpp)**.

- Use a static C-style string (character array) to store the string.

- Be sure to initialize your C-style string before it is used.
- Use `cin.getline()` to read in the user's string.
- Use `strlen()` to get the length of the string.
- Run your test cases and check that they work correctly.

(2 pts) D. Passing Arrays to Functions

1. Rewrite the previous program to use a function to compute the star-factor of a string **(lab7_star_func.cpp)**. The function declaration should be:

```
float star_factor(char* str);
```

2. Inside `main()`, call your function with the user's string (which is stored in a static array), then create a new string (off the heap) that is large enough to store twice as many characters as the first string. Use `strcat()` to duplicate the string (if the user entered "`look*up`" then the new string should be "`look*uplook*up`"). Call your `star_factor()` function with the new string and print the result. Is it the same or different than for the first string?

Be sure to delete the new string in `main()` when you are done using it.

3. (Optional) If you have extra time, write a new function that takes in a string and a desired star-factor value, then returns a new string that achieves at least the desired star-factor (by adding '`*`' characters as needed). You will need to compute how many additional characters are needed and allocate enough space to store the new string and all of its characters.

```
char* star_percent(char* str, float desired_star_factor);
```

Ask a TA to check your work (get points) and submit your programs to TEACH

1. Transfer your .cpp file from the ENGR servers to your local laptop.
2. Connect to TEACH here: https://teach.engr.oregonstate.edu/teach.php
3. In the menu on the right side, go to **Class Tools -> Submit Assignment.**
4. Select **CS 161 020 Lab_7** from the list of assignments and click “SUBMIT NOW”.
5. Select your .cpp files (`lab7_fruitfly.cpp, lab7_rainfall.cpp, lab7_star.cpp, lab7_star_func.cpp`).
6. Click the **Submit** button.
7. You are done!

Point totals: 3 pts (quiz) + 2 pts (arrays) + 3 pts (C-style strings) + 2 pts (passing arrays)

If you finish the lab early, this is a chance to work on your Assignment 4 implementation (with TAs nearby to answer questions!).
