COMP30820 Mini-Assignment 1

Last Updated: February 2021

Submission Details

Deadline: Feb 12th 2021, 5pm (Uploaded to Brightspace).

Late Submissions Standard UCD policy on late submissions applies; see `https://www.ucd.ie/t4cms/latesub_po.pdf`. Your submission is deemed late if at least one deliverable is submitted late.

Plagiarism: The submission must be yours and yours alone. If you are unsure what is or is not plagiarism, the following is a none exhaustive list of example activities you cannot do:

- Copy the completed files of another student and submit them as your own
- Share copies, images or print outs of your code with another student (by e-mail, FB messenger, WhatsApp etc.)
- A group of students working on a single solution and then all submitting the same work or subset of the same work (regardless of whether variable, method, class names or ordering have been changed)
- Students collaborating at too detailed a level. For example, consulting each other after each line / block / segment of code and/or sharing the results.

For more details see:

- `https://csintranet.ucd.ie/sites/default/files/cs-plagiarism-policy_sept2020.pdf`, and
- `https://www.ucd.ie/governance/resources/policypage-plagiarismpolicy/`
- `https://www.ucd.ie/secca/studentconduct/`

Problems

1. Given an airplane's acceleration a and take-off speed v, you can compute the minimum runway *length* needed for take-off on a dry runway using the following formula:

   $$length = \frac{v^2}{2a} \tag{1}$$

   Write a program that promps the user to enter a number v for speed, a number a for acceleration, and a character 'w' or 'd' to indicate if the runway is wet or dry. Then, display the calcalated runway length with the above formula, increasing the result of the formula by 15% if the runway is wet.

   The third input should accept lower or upper case w or d and exit with an error message on any other input.

2. Write a program that prompts the user to enter the year as an integer, and the month number as an integer from 1–12, and displays the number of days in the month. For example, if the year number is 2020, and the month number is 2, the program should display "29 days". The program should return an error message if the month number is not between 1–12.

3. Write a program that randomly generates an integer between 0 and 100, inclusive. The program prompts the user to enter a number continuously until the number matches the randomly generated number. For each user input, the program tells the user whether the number is too low or too high, for example, if the randomly generated integer were 81, a program run might look like:

```
Guess a number from 0-100.
Enter a number:
>50
Too low.
Enter a number:
>75
Too low.
>87
Too high.
>81
Correct!
```

4. Write a program that prompts the user to enter two strings and displays the largest common prefix of the two strings. Here are some examples (note that case should be taken into account when finding prefixes; for example, the characters ’a’ and ’A’ should not be considered a match):

   For input strings “Car” and “Cars”, the program should display ”Car”

   For input strings “Welcome to Java!” and “Welcome to Java”, the program should display “Welcome to Java”

   For input strings “Hi ALL” and “Hi All”, the program should display “Hi A”

   For input strings “Abc” and “abc”, the program should display “no common prefix”

Deliverable

Code All .java files as a .zip archive, one file for each problem.

Marking Guidelines

Each problem will be marked out of 15, with 10 marks for Specification and 5 for Style.

Criteria	**A+, A, A-**	**B+, B, B-**	**C+, C, C-**	**D+, D, D-**	**≤ E+**
Specification (10/15)	Program works perfectly with required input validation.	Program works perfectly on expected input, logic error or runtime error on unexpected input	Minor logic errors, or logic errors in one part of the program flow	Program compiles and runs but lacks major functionality	Program does not compile, or performs little to none of the required behaviour
Style (5/15)	The code is easy to read, appropriately commented, and uses an efficient solution	Some minor faults in presentation, such as inadequate comments, poor variable naming, or a slightly awkward approach	Some style problems throughout the comments, variable names, and choice of implementation	A major problem in implementation (e.g. repeating multiple statements when a loop should have been used), or naming or commenting that makes the code very difficult to read.	Major problems in all of the above.