CCPS506 Lab 7 – Rust is Cool (and Tedious)

Preamble

In this lab you will experiment with writing some simple Rust functions. This would be CCPS109-level stuff in Java but passing arguments and returning values in Rust becomes a lot trickier due to issues of ownership and strict typing.

Lab Description

Implement each of the following Rust functions in the single file called lab7.rs

i) Write a Rust function called **`stringConcat`** that accepts two Strings (or string slices) as input and returns the concatenation of the two.

ii) Write a Rust function called **`subArrayAverage`** that accepts an array of floating-point values, and a tuple containing an upper and lower bound. Return the average of the values in the sub-array defined by the upper and lower bounds. The lower bound is inclusive, the upper bound is not. Assume the tuple contains valid bounds. No need to error check.

iii) Write a Rust function called **`arraySignum`**. It should accept an array of integers, and return an array containing the values -1, 0, or 1 depending on the sign of the elements in the argument array. Your function should return a new array. Do not overwrite the input array.

In addition to these three functions, your lab7.rs file should contain a main function that calls each of the above functions with an example or two to make sure they work correctly.

Submission

Place your implementations for the above functions in a single file called **lab7.rs**

Labs are to be submitted ***individually***! Submit lab7.rs on D2L.