**CS 3353 – Spring 2021 - Midterm Take Home Exam**

*Distributed: Friday March 26, 2021: Submission Due: Friday April 2, 2021*

Technical mastery of foundational computer science topics is important for any CS major. Just as important is being able to communicate technical content in a clear and concise manner. This “midterm” will exercise both skills.

Here is what you will do:

1. Choose one of the following for your Algorithm Adventure.
   1. A data structure that is new to you. Specifically, it should not be one covered in a previous class. There is no way for me to monitor whether you have come to know the data structure in a non-class context, so I am taking your word that it is new to you.
   2. An algorithm that we have not covered in class and that you have not covered in some other class. Same caveat as in 1.a about taking you at your word.
   3. A container class or algorithm implemented in Facebook’s Folly library. The Folly Library makes some bold assertions about performance of its containers and algorithms when compared to the c++ std library. For this option, you could choose the Folly equivalent of something you are familiar from cpp std lib.
2. Learn about it.
   1. What problem does it address with its predecessors or, in the case of 1.c, what issue(s) does the Folly class attempt to improve upon from std lib?
   2. Learn how it works.
      1. For a data structure, you need to become familiar with the fundamental operations such as insert, search, remove, etc.
      2. For an algorithm, you need to teach yourself how it works.
      3. For a Folly container/algo, you’ll need to pour over the docs and source code and understand what improvements Facebook made over std lib.
   3. Code up some examples to confirm your understanding.
3. Compose an SMUCS Medium.com (<https://medium.com/smucs>) blog post describing your data structure, algorithm, or Folly container/algo. It should include:
   1. A brief history or motivating example for why the new thing was needed.
   2. An explanation with examples and graphics about how your selection functions/operate and how it improves on the previous options or equivalent in std lib. You can use pseudocode in this example to explain the process, but don’t start off with a big chunk of code.
   3. A walkthrough of some code examples demonstrating features, functionality, benefits of your selection.
      1. You don’t necessarily need to code from scratch here, but you should definitely be the one who writes tests or the examples. For example, if your algorithm already exists in Boost, your examples can be using the Boost implementation.
      2. The examples do no need to be in C++ (except in the case of 1.c Folly since it is a c++ library).
   4. A “Further Reading” Section and a References cited section.
   5. A link to a public Github repo of your example code. Remember, even though it is for example code to deepen understanding of your selected ds/algo/library, it needs to be well documented and professional.

**FAQ**:

Q: How do I write a post for the SMU CS Tech Medium Publication?

A: Dr. Fontenot will add all the students in the class as authors for the publication ASAP.

Q: How long does the write up need to be?

A: The write up should be as long as needed to adequately communicate your new-found knowledge, but no longer.

Q: How many examples do I need?

A: More than 1, less than 100.

Q: How do I write a Medium post?

A: Check out this medium post: <https://medium.com/@codytowstik/medium-post-formatting-guide-77e9f8307d32>

Q: How do I make graphics for my post?

A: A solid way to do this is Google Draw. You can export the graphics in a number of different formats. Be careful about bitmapped graphics as they will get blurry if you resize them. There are tons of other options out there, though.

Possibly more suggestions/thoughts to come …