Program-Level L(ch)-6.1 (2016 S͏e͏h͏o͏o͏l͏3 of ͏S͏c͏i͏e͏n͏c͏e͏ ͏a͏n͏d͏ ͏E͏n͏g͏i͏n͏e͏e͏r͏i͏n͏g͏ic4 Tw at

Date (Month/Year): Assessment Contact: Erin Chambers

In what year was the data upon which this report is based collected? AY 2021-2022

In what year was the program's assessment plan most recently reviewed/updated? 2018

Is this program accredited by an external program/disciplinary/specialized accrediting organization or subject to

lyes pesesore th isrecits t1.3hrne programs8

articles; note this might be a challenge, for reasons I'm happy to discuss. Or come see me if you have other creative ideas! The idea is to show significant knowledge on some advanced data structures topic, but I'm prepared to be flexible for this category.

CSCI 5300: Students were asked to participate in an in-class assessment and were rewarded with a small participation credit. Students were instructed to not use any internet resources to answer assessment questions. Since the grade was based on completeness and not correctness, most of the answers likely represent students' learning outcomes.

Assessment details:

1. Pick one data structure you are using in your team project. Briefly describe the data structure and the purpose it serves in your project.
2. Analyze the choice of this data structure for the purpose it serves in terms of program efficiency, coupling, and/or cohesion.
3. What alternative data structure could you have used? Analyze if this alternative would be a better choice for your project.
4. Explain what the term “security” means in the context of software.
5. Describe what measures you would take to ensure that the software you produce is “secure”.
6. State and explain what you believe is the ideal team size for a: Small project (about the size of our class project) and a medium project
7. Given your ideal team size and project requirements, explain how you would organize your team and approach the development process to deliver the required software.
8. Describe the git workflow we have utilized in this class for the team project.
9. Explain the difference between the workflow we used in this class and the workflows you have used in other situations.
10. What considerations do you need to take into account when deciding on what workflow to use?
11. Explain how we applied various development tools to assure code quality.

2	6	4
1	9	5

For the final project:

Score	Data Structures	Algorithms
4	4	6
3	3	6

example, perhaps you’ve initiated one or more of the following:

Changes to the Curriculum or Pedagogies

- Course content
- Teaching techniques
- Improvements in technology
- Prerequisites
- Course sequence
- New courses
- Deletion of courses
- Changes in frequency or scheduling of course offerings

Changes to the Assessment Plan

- Student learning outcomes
- Artifacts of student learning
- Evaluation process
- Evaluation tools (e.g., rubrics)
- Data collection methods
- Frequency of data collection

Please describe the actions you are taking as a result of these findings.

As a result of both this year’

D. How do you plan to (continue to) use this information moving forward?

N/A

IMPORTANT: Please submit any assessment tools (e.g., artifact prompts, rubrics) with this report as separate attachments or copied and pasted/appended into this Word document. Please do not just refer to the assessment plan; the report should serve as a stand-alone document. Thank you.

PLO 3 - Application of Theory, Systems, and Software Development Fundamentals

Outcomes

Graduates of the program will have an ability to...

BA-CS, BS-CS, MS-CS

Application of Computer Systems Fundamentals

Criterion	4: Exemplary	3: Accomplished	2: Developing	1: Beginning
Program Execution	Student can **critically evaluate** execution management strategies in real contexts and **adapt or create** new strategies to accomplish or optimize system goals.	Student can **implement or describe a concrete implementation** of different code execution strategies to achieve desired system-level outcomes.	Student can **reason about** how and when a system executes code to accomplish its goals. Students can **compare and contrast** different systems and explain why they manage code execution differently.	Student can **describe** how programs, processes, threads, tasklets, or other runnable code is executed on hardware in an abstract, idealized manner. Student can **describe** mechanisms and algorithms that manage computing time as a resource.
Memory and Data Mangement	Student can **critically evaluate** data management strategies in real contexts and **adapt or create** new strategies to accomplish or optimize system goals.	Student can **implement or describe a concrete implementation** of different data management strategies to achieve desired system-level outcomes.	Student can **reason about** how a system manages data storage and movement to accomplish its goals. Students can **compare and contrast** different systems and explain why they manage data differently.	Student can **describe** how data management systems (memory, cache, databases, etc.) function in an abstract, idealized manner. Student can **describe** how computer data is managed as a resource.
Networking	Student can **critically evaluate** networking strategies in real contexts and **adapt or create** new strategies to accomplish or optimize system goals.	Student can **implement or describe a concrete implementation** of different networked communication strategies to achieve desired system-level outcomes.	Student can **reason about** how distributed systems use communication to accomplish their goals. Student can **compare and contrast** different systems and explain why they manage communication differently.	Student can **describe** how network hardware and software operates in an abstract, idealized manner. Student can **describe** protocols and algorithms that manage the transfer of information between systems.
Security				

Notes on the above rubric

- This learning outcome evaluates the students' process of applying learned knowledge and skills to a specific problem, not necessarily the specific skills and learned knowledge itself.
- PLO3 is a broad learning outcome that applies to many courses. This rubric attempts to be general enough so that elements may be applicable to any course covered under PLO3. It is not intended to be specific to the Computer Systems courses. For example, the Algorithms course could incorporate elements of "Program Execution" by analyzing an algorithm's Big-O running time under two models: one where a single instruction occurs per time step (sequential execution) versus another where all possible instructions occur per time step (infinitely parallel execution). Or, the Algorithms course could incorporate elements of "Memory and Data Management" by discussing working-set-size and in-cache versus out-of-cache algorithms or in-core and out-of-core algorithms.
- This rubric attempts to hit Computer Systems concerns at a high and low level. For "Memory and Data Management" a programming course may talk about how the Java garbage collector manages memory, an architecture course may talk about how the CPU cache interacts with memory, an OS course may talk about virtual memory and paging, a database course may talk about database organization, and a security course may talk about where data is encrypted and decrypted.
- In many courses these four dimensions of computer systems will interrelate to one another, even if there are apparently one or two primary dimensions. For example, a networking or distributed systems course might talk about efficiently distributing computation and data storage across client and server, subject to the security concerns of who is trusted to do what kinds of operations.

Application of Software Development Fundamentals

Criterion	4: Exemplary	3: Accomplished	2: Developing	1: Beginning
Team and Work Organization	Student can critically evaluate software development strategies in real contexts and adapt or create new strate-			