CSCI 3901 Lab 1: Basic Problem Solving

Fall 2024

Objective

In this lab, you will practice the process of problem solving and coding when you may not have all of the details of the original problem. For some, this will seem easy. For others, there will be some challenge to grasp everything within the lab time.

Working in groups of one or two students, you will write code to implement a Java Map.

In Java, a Map is a structure that stores pairs of values, one as the key and the other as the value. You provide the Map with the key and the Map returns the value stored with that key.

In addition to a constructor, you should implement the methods "get", "put", "size", and "contains" for your Map. Do not worry about other Map methods.

In your implementation, you cannot use any of the existing Java Set or Map classes.

Preparation

Ensure that you have your Integrated Development Editor (IDE) for Java installed (classroom homework).

Resources

No external resources needed to bring in.

Procedure

Set-up

None

Lab steps

1. Ensure that you and your partner agree on an understanding of the problem.
2. List the elements of the problem statement that are not defined sufficiently-well to begin an implementation.
3. Select a resolution for the missing parts of step 2.
4. Decide on the approach you will use to implement the methods
    - How you will store the data
    - What difficulties you will expect
5. Implement your solution.
    - You are permitted to use an AI tool to produce code if you want. In that instance, you must properly cite the use and provide the exchange that

generated the code. In such a case, you are still responsible for the quality and the correctness of the code.

6. Update any assumptions made in the implementation that you didn't identify earlier, clearly distinguishing these latter assumptions from those you made before starting the code.
    - If you used an AI tool to generate the code, you are still responsible for identifying the assumptions made by the code towards the problem solution. Using a tool does not absolve you of understanding what the tool has created for you.

Analysis

1. Identify how you will know that your implementation is working.
2. Identify what assurance we have that the code has some quality to it (particularly important if you had an AI tool generate the code)
3. What difficulties, if any, did you encounter in finally obtaining some working code, and how you dealt with the difficulties.
4. **Briefly** describe either
    a. How you would approach this problem differently if you had to start new; or
    b. What you did well in developing the implementation that you could use as an approach to coding a solution to another problem.

Reporting

1. In one file, list
    - Your team members
    - The list of items from the description that needed clarification
    - Your decisions on the items that need clarification
    - How you showed that your work (so far) is working
2. In that same file, include the answers to the questions from the analysis part of the lab.
3. Generate a PDF from the document.
4. Submit the PDF and your Java files in Brightspace in the Lab/Lab 1 section of the course page in Brightspace by Friday, September 13 at 11:59pm Halifax time. **Note:** Only one member of the group needs to submit the report.

Assessment

The assessment will be on a letter grade and will reflect how well you are demonstrating and executing on problem solving techniques that could be used in other situations. I am more concerned with your thinking about how to get a solution than on the code being complete solution.

Grading scheme

Letter grade	Items that need clarification	Decisions on items for clarification	Implementation	Demonstration of work so far	Assessment of quality and challenges	Looking forward
A (Excellent)	Have thought through all the aspects of the problem	Resolution for unclear items are well thought-out based on future direction of the work	The Map implementation is complete and well-done. Well documented and easy to build on later. Guards against input validation problems.	There is a definite plan on how to show that the code works. All the cases are covered.	Understands what comprises quality code and can self-assess objectively. Understands a strategy for getting through difficulties.	Have a strategy that will carry forward to other problems.
B (Good)	List of items is relatively complete	Reasonable conclusions made about each unclear item, along with signs of thinking through the conclusions.	The Map implementation is complete and has all the relevant functionality working.	There is some notion of what's needed to show the code is working. May not be fully implemented here.	Understands what comprises quality code, can reflect somewhat on their own code, and can explain the nature and specifics of difficulties encountered.	Have identified what worked or should have worked in this problem.

C (Satisfactory)	There are enough key items identified to get started, but more will arise as you proceed	There is a consistent set of conclusions made for the items.	General framework of the implementation is in place and the major components are done.	Has a notion of using testing on the code. Incomplete set of tests.	Understands elements of quality code and can identify where difficulties in development arose.	Show an awareness that some things might make it easier or harder, but may not be able to identify them clearly.
D (Marginal Pass)	Approach seems to be to find the items as you go	There is some resolution to the items, though some may be inconsistent or will lead to problems later	At least one of the methods is in place and on track to be working.	Provide some argument with some basis on why the code is working. Nothing formal or complete.	Marginal awareness of what is quality code or development challenges.	Minimal awareness that the strategy by which we approach a problem can make a difference on the solution.
F (Inadequate)	No discussion on items to clarify	Lack of clarity on the resolution	Little code is done and/or is not in a state to work.	Little convincing argument beyond an	Unaware of what constitutes quality or	Little awareness of what can make

		of the items.		assertion that the code works.	unaware on how you got through difficulties other than an attitude of “well, we got code”	the problem solving process go well.