**Lab Assignment #6 – Using Maps and Hash Tables**

Due Date: Friday, Week 11

Purpose: The purpose of this Lab assignment is to:

1. Design algorithms that describe operations on ADT Maps
2. Investigate the efficiency of hash table implementations
3. Implement and test appropriate methods in Java or Python

References: Read the course’s text chapter 10 and the lecture slides. This material provides the necessary information that you need to complete the exercises.

Be sure to read the following general instructions carefully:

- This assignment must be completed individually by all the students.

- See the naming and submission rules at the end of this document

**Exercise 1**

**If your first name starts with a letter from A-J inclusively:**

Our *AbstractHashMap* class maintains a load factor l ≤ 0.5. Reimplement that class to allow the user to specify the maximum load, and adjust the concrete subclasses accordingly.

Perform experiments on our *ProbeHashMap* classes to measure its efficiency using random key sets and varying limits on the load factor. Do you think ProbeHashMap is better or ChainHashMap? When and how?

**Hint** The load factor can be controlled from within the abstract class, but there must be means for setting the parameter (either through the constructor, or a new method).

Write a Java/Python application to test your solution.

**If your first name starts with a letter from K-Z inclusively:**

Our *AbstractHashMap* class maintains a load factor l ≤ 0.5. Reimplement that class to allow the user to specify the maximum load, and adjust the concrete subclasses accordingly.

Perform experiments on our *ChainHashMap* classes to measure its efficiency using random key sets and varying limits on the load factor. Do you think ProbeHashMap is better or ChainHashMap? When and how?

**Hint** The load factor can be controlled from within the abstract class, but there must be means for setting the parameter (either through the constructor, or a new method).

Write a Java/Python application to test your solution (7 marks)

**Exercise 2**

**If your first name starts with a letter from A-J inclusively:**

The use of **null** values in a map is problematic, as there is then no way to differentiate whether a **null** value returned by the call get(k) represents the legitimate value of an entry (k, null), or designates that key k was not found. The java.util.Map interface includes a boolean method, *containsKey(k)*, that resolves any such ambiguity.

Implement the *containKey(k)* method for the *SortedTableMap* class. **Hint:** Use the existing *findIndex(k)* method.

Write a Java/Python application to test your solution.

**If your first name starts with a letter from K-Z inclusively:**

Implement the *containKey(k)* method for the *UnSortedTableMap* class. **Hint:** Use the existing *findIndex(k)* method.

Write a Java/Python application to test your solution.

(3 marks)

**Evaluation:**

|  |  |
| --- | --- |
| **Functionality:**   * Correct implementation of requirements * Code demonstration and brief explanation in a short video | 70%  10% |
| **Object-Oriented design**:   * Correct design of classes and methods similarly to chapter 3 examples. * Correct use of generics * Correct use of naming guidelines for classes, variables, methods. | 15%  5% |
| **Total** | 100% |

**Naming and Submission Rules:**

You must **name your Eclipse project** according to the following rule:

**YourFullname\_COMP254Labnumber**. Example: **JohnSmith\_COMP254Lab1**

You must name package names **com.exercisenumber.yourfirstname.yourlastname**, for example: com.exercise1.john.smith

Provide your **student number and full name as a comment** at the top of main method for each exercise.

**Archive your project in a zip file** named according to the following rule:

**YourFullname\_COMP254Labnumber.zip**

Example: **JohnSmith\_COMP254Lab1.zip**

Upload the zip file on eCentennial using the Assignment link.