

**CARLETON UNIVERSITY**  
**Department of Systems and Computer Engineering**  
**SYSC 2100 Algorithms and Data Structures**  
**Queues and Trees**

Assignment 5

Due: **Sunday, April 2<sup>nd</sup> 2017, 11:00pm**

---

This assignment consists of two parts:

1. A Java compiler uses a symbol table to keep track of the identifiers that a program uses. When the compiler encounters an identifier, it searches the symbol table to see whether that identifier has already been encountered. If the identifier is new, it is inserted into the table. Thus, the symbol table needs only insertion and retrieval operations.

Write an implementation of a symbol table that uses hashing. Use the hash function  $h(x) = x \bmod \text{tableSize}$  and the algorithm that involves Homer's rule, as described in the section "Hash Functions," in the textbook to convert a variable into an integer  $x$ . Resolve collisions by using separate chaining. (See the code given on pages 748 to 750 of Chapter 13.) Since you add an item to the table only if it is not already present, does the time required for an insertion increase?

In the main line of the program provide test code that inserts symbols such as "foo", "bar" and "baz". Also ensure that your test code handles collisions and that it attempts to retrieve symbols which were not inserted.

2. Implement the ADT table described in Chapter 13 of the text book using a **2-3 tree**. For simplicity, you can assume that the table only stores integers and that the key equals the value (i.e, you are implementing a set using a 2-3 tree).

In the main line of the program provide test code that inserts 1 2 3 4 5 6 7 8 9 10, prints the table, then deletes 3 and 7, prints the table, and finally deletes 13.

**Submission Requirements:** Submit your assignment (the source files) using cuLearn. Your program should compile and run as is in the default lab environment, and the code should be well documented. Submit all files without using any archive or compression as separate files. The main program for part 1 should be called `HashTable.java` and for part 2, your submission should be called `TwoThreeTree.java`. If you need to define additional classes etc., you are free to name them according to your own needs. But the TA(s) should be able to run your application by entering `java HashTable` and `java TwoThreeTree` on a command-line.

Marks will be based on:

	<b>25%</b>	<b>50%</b>	<b>75%</b>	<b>100%</b>
<b>Part 1 (6 marks)</b>	Compiles	Runs	Inserts and prints correctly.	Handles errors gracefully.
<b>Part 2 (8 marks)</b>	Runs	Inserts and prints correctly.	Deletes items in list correctly.	Handles errors gracefully.
<b>Readability (3 marks)</b>	Hard to decipher.	Dead or useless code or variables, reinventing wheel, methods that do too much	Follows conventions, proper whitespace and indenting, appropriate and meaningful naming of variables and methods.	Includes logging or testing code for tree insert and delete (commented out or disabled for submission)
<b>Comments (2 marks)</b>	Occasional	Some	Lots, but extraneous (i.e, <code>i=j //copy j to i.</code> )	Javadoc comments for all member functions, classes. Minimal and cohesive comments elsewhere.
<b>Completeness of your submission (1 mark)</b>	Something in CULearn	TA must change code to get it to run.	RAR, JAR, ZIP files, incorrect naming.	Strictly adheres to the submission requirements.

The due date is based on the time of the cuLearn server and will be strictly enforced. If you are concerned about missing the deadline, here is a tip: multiple submissions are allowed. So you can always submit a (partial) solution early, and resubmit an improved solution later. This way, you will reduce the risk of running late, for whatever reason (slow computers/networks, unsynchronized clocks, failure of the Internet connection at home, etc.).

In cuLearn, you can manage the submission until the deadline, taking it back, deleting/adding files, etc, and resubmitting it. The system also provides online feedback whether you submitted something for an assignment. It may take a while to learn the submission process, so I would encourage you to experiment with it early and contact the TA(s) in case you have problems, as only assignments properly and timely submitted using cuLearn will be marked and will earn you assignment credits.