

# KIT205 Data Structures and Algorithms

## Week 5 Tutorial

In this tutorial you will be implementing a hash table using separate chaining. The separate chaining will make use of the linked list data structure that you implemented in week 2. However, you will be storing strings in your hash table, so the list will need to be modified to store strings. A new project is provided to get you started.

## Hash Table Implementation

1. Download *TutorialWk5.zip* and unzip to your working directory
2. Open the project and modify the *list.h* and *list.c* files so that the list stores strings instead of ints. There are some hints in the comments. The most important point is that you should allocate just enough memory for the string to be added (you will use a similar technique for the assignment). You will also need to make use of the string functions *strlen*, *strcpy*, and *strcmp*. You can find documentation for these functions here: [https://en.wikibooks.org/wiki/C\\_Programming/Strings](https://en.wikibooks.org/wiki/C_Programming/Strings)
3. Open *hashtable.h* and familiarise yourself with the hash table interface.
4. Now open *hashtable.c*. You will see that some functions have been implemented and some have not. Make sure that you understand the existing functions.
5. Now implement *create\_hashtable*. This function should create an array of lists of length *size*. Each list needs to be initialised to *NULL*.
6. Next implement *hash\_insert*. This function will simply call *insert\_at\_front* after determining which list to insert into using the *hash* function (make sure you pass a *pointer* to the list).
7. Open the *week5.c* file and familiarise yourself with the main program loop (this will also be similar to your assignment loop).
8. Add some code to case 3 that simply calls the *hash\_print* function.
9. Test your work so far by running the program, inserting some strings, and printing the resulting hash table.
10. If you have time, implement and test the remaining functions