# CSci 1933 Lab 10

November 17, 2015

## 1. Introduction

The purpose of this lab is to familiarize you with the creation and use of Java's built in Map, specifically the HashMap implementation.

## 2. Task 1

In today's lab, you will implement a simple translator which works on text replacement. In general, what this means is that your implementation will replace the english input string with the matched word in the provided data file. You will use a Map to do this task.

To complete this lab, you will do the following:

- In IntelliJ, create a new Java project named lab10.
- Import the following files into your lab10 project.
  - DictionaryReader.java This class will read in associated words from a dictionary file. Place this in the src/
  - Dictionary.txt A dictionary to translate from English into Romanian. Place this in your project root.
- Create a Translator class, with a main() method.
- Pass command line arguments to the program:
  - Click on run menu.
  - Open the "Edit Configurations..." dialog box, go to Defaults->Application.
  - In the 'Program Arguments' input box, and enter the sentence "where is a good place to get coffee".
- In your main() method, check the size of the array of passed in arguments (the String[]args parameter). If no arguments have been passed in, print an error message and exit. To exit a Java program before the end of the

main() method, call System.exit(), with an exit code. Since this is an error, we want to exit with a value of '1'.

```
System.exit(1);
```

- Create a Map where the keys and values are both of type String. The template of the Map generic is:

```
Map<key, value> variable = new HashMap<key, value>()
```

- Create a DictionaryReader object by passing in the path ./Dictionary.txt to the constructor, similar to the way you have used TweetReader in the past.
- Read in the entries using DictionaryReader's getKey() and getValue(), then place them in the Map using:

```
map.put(key, value)
```

- Iterate through the elements of the args array
  - Check if the current element is present in the Map using Map's containsKey(key) method.
  - If it is present then find its corresponding value in Romanian (use map.get(key)) and append it to the translated sentence. The following code segment appends the string " World" to "Hello". Remember to put spaces between the translated words.

```
String sentence = "";
sentence += "Hello";
sentence += " " +"World";
```

  - If it isn't present then print an error message and exit.
- Run the program, the translated output should print in the console

## 3. Task 2

In last task, we used JAVA's inbuilt HashMap implementation. Now we'll implement a very basic HashMap ourselves using associative arrays.

- Create a new class KeyValuePair. It should have two String variables key and value. The constructor should be able to initialize these values.
- Create another class MyHashMap. Declare a linked-list of KeyValuePairs in this class. Now we'll implement some primary routines in following steps.
- containsKey(key): This function should check if a key is present in our map, by iterating the through the linked list and return a boolean value depending upon that.
- put(key, value): This function should first check if a key is already present in our map. If it is present, just change its value to our method argument. Otherwise, insert a KeyValuePair (initialized to method arguments) to the linked-list of KeyValuePairs.
- get(key): This function should iterate the linked-list of .KeyValuePairs to find if we can find a matching key. If we succeed, we just return its value. If no such key is present, we return NULL.
- Now, go back to your Translator class and modify its main function to use your newly implemented MyHashMap instead of JAVA's HashMap. Confirm that you get the same translation as before.

## 4. Extras

You can complete this part if you still have time.

- Implement the remove(key) method. Check if the linked-list contains key. If yes, remove the corresponding key-value pair.
- Implement the keyList() method. It should return the list of keys currently present in our map.
- Think of changes we require if we use ArrayList instead of LinkedList to store key-value pairs. What are the pros and cons of using ArrayList? Think in terms of operations we need from MyHashMap.

**------Solution will be available in last 15 mins------**