

# WIIT-7320 Java II – Lab 4

## Synopsis

The US Census bureau conducts a census of the US population every 10 years. The Census bureau has a document available on the web listing over 80,000 common surnames ranked in order of frequency.

The URL of the document is:

<https://www2.census.gov/topics/genealogy/1990surnames/dist.all.last>

The document contains four fields. The first is the surname. The second is the frequency of the surname expressed as a percentage. The third field contains a cumulative frequency, also expressed as a percentage. The final field is the ranking of the surname. For example 'SMITH' is the most common surname in the United States, hence it is ranked as number 1.

Here is a listing of the first ten rows in the file.

SMITH	1.006	1.006	1
JOHNSON	0.810	1.816	2
WILLIAMS	0.699	2.515	3
JONES	0.621	3.136	4
BROWN	0.621	3.757	5
DAVIS	0.480	4.237	6
MILLER	0.424	4.660	7
WILSON	0.339	5.000	8
MOORE	0.312	5.312	9
TAYLOR	0.311	5.623	10

You will write two programs, one that reads this data from the web and stores it locally. The next that looks up a surname from the list and prints frequency and rank.

## Part 1

Create a Java class called Surname that holds a surname, the frequency of that surname (**not** cumulative frequency), and the rank of the surname.

Create a Java class with a main method called CensusData. This class will read data from the URL above. It will use each row of the Census data file to instantiate a Surname object. That Surname object will then be added to a HashMap collection using the surname string as a key.

Finally, after the HashMap is fully populated, it will be serialized into a local file named `./surname.dat`.

Implement and test your code.

## Part 2

Create a Java class with a main method called Lookup. This class will deserialize the previously created file to re-create the HashMap containing the Surname objects. The Lookup class will then continuously query the user to enter a surname string. If the user supplied surname is in the HashMap, the program

will print the surname, the surname frequency, and the surname rank. Otherwise it will inform the user that the surname is not in the list. The query should be case insensitive. The program should terminate when the user enters 'quit'.

## Sample Output

```
Enter a surname (or quit to end): Smith
Surname: SMITH frequency: 1.006% rank: 1
Enter a surname (or quit to end): McNally
Surname: MCNALLY frequency: 0.007% rank: 1795
Enter a surname (or quit to end): Mohammed
Surname: MOHAMMED frequency: 0.002% rank: 6531
Enter a surname (or quit to end): Palzer
Surname: Palzer not found
Enter a surname (or quit to end): quit
```

Process finished with exit code 0