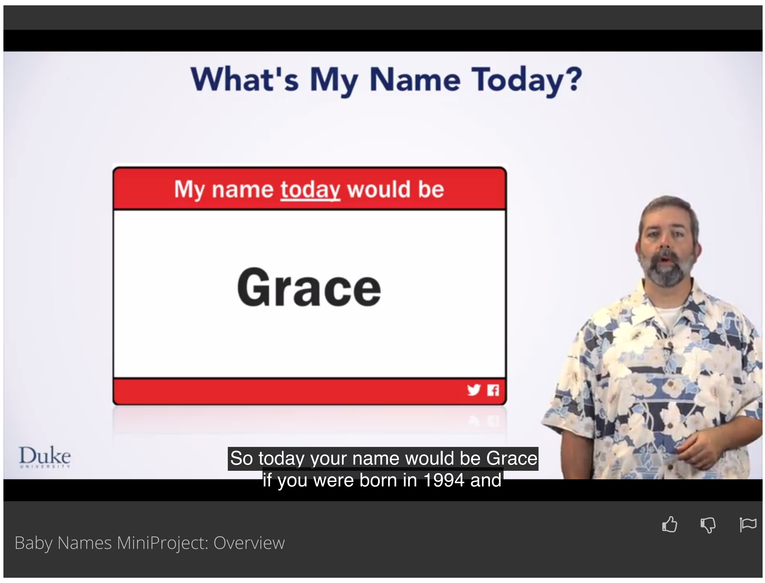
Walkthrough

Batch Grayscale Images



Baby Names MiniProject: Overview

Have a question? Discuss this lecture in the week forums.

Interactive Transcript

Search Transcript

English

0:03
Hello. You're going to learn the background and coding needed for the mini project for this course. Your going to write programs to answer questions that are very hard to answer using a spreadsheet, but that you'll be able to approach in a straightforward way using the practices, skills, and libraries you've learned about in this course. You're going to answer the question, what's your name this year, giving your name and the year you were born. In other words, what name this year has the same popularity rank as your name did the year you were born. You can answer this question for anyone's name. So you can use a friend's name, a singer you like, or anyone. These pictures we will be using to describe what you're going to be doing are taken from a website that offers something similar to what you'll be doing in this mini project.

0:48
Suppose you are a female named Jennifer born in 1994. Jennifer was the 21st most popular girl's name in 1994. So one task you'll need to do is write code to figure out a name's ranking for a given year.

1:04
If you were Jennifer in 1994, what would your name be today?

1:09
You'll discover via coding that Grace is the 21st most popular name today, where today is 2014, the most recent year for which we have data on names given to babies in the United States.

1:22
So today your name would be Grace if you were born in 1994 and named Jennifer, but you can also see what your name would be in any given year.

1:32
Here we see a summary showing your name in several different decades. Jennifer in 1994 would be Barbara in the 1970s.

1:40
That means Barbara was the 21st most popular name

1:44
in the decade of the 1970s when taken together.

1:48
Because we have data for the United States going back to the 1880s, more than 130 years ago, we can determine your name going back a long time.

1:58
Here we see that your name in the 1900s would be Sarah if you were Jennifer in 1994. You'll need to write programs to make conclusions about the names that we've outlined here, turning all that data into information.

2:13
The US Government releases baby-name data every year. We've collected that data and made it available to you as part of this course. We would be excited to have your help to collect similar data for other countries in the world.

2:26
We have data for males and females going back many years with a different data file for every year.

2:34
The files share a naming convention, which is convenient when writing programs to open the files and read them. You'll leverage the common naming convention in writing code to access these hundreds of data files.

2:46
The file contents are also similarly formatted. That will help you write more general code to solve these problems. We'll look briefly at the data file for 2014, the most recent data provided to you for use.

3:00
The line numbers in the file are ordered by the number of babies with a given name. So the most popular name, baby name, comes first, then the second most popular, and so on. As you can see here, 20,799 babies were named Emma in 2014, making it the most popular female baby name that year.

3:22
All of the female names precede the male names in the data file. This means the most popular boy's name, which was Noah in 2014, with 19,144 boys named Noah, comes just after the girl names that the fewest girls have, which were Zyrihanna and Zyriyah in 2014. We'll walk through the high-level concepts for accessing data in one file, but you'll need to go through the full seven steps to solve the problem we're asking you to address in this mini project. You'll use several classes from the edu.duke and org.apache.commons packages in developing your solution to this problem. For example, you'll need a file resource object to access the data in a file for a particular year. You'll need a CSV parser by calling getCSVParser method from the FileResource class, making sure that you ask for a parser with no header row.

4:16
Because there is no header row, you'll need to access the data in each record by indexing. The first data element with index 0 is the baby's name in the file.

4:27
The gender of the baby is the second data element. Once you've got this, you'll be ready to start thinking about the problem and using our seven-step process to solve the entire problem. Have fun.

Downloads

Lecture Video mp4
Subtitles (English) WebVTT
Transcript (English) txt
Lecture Slides pdf

Would you like to help us translate the transcript and subtitles into additional languages?