

Leslie Flores  
20077795  
CMPT 440L 111 20S  
Formal Language Theory  
January 27<sup>th</sup>, 2020  
Professor Rivas  
Homework 0

```
Command Prompt
Microsoft Windows [Version 10.0.17763.973]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\lesli>java -version
java version "1.8.0_241"
Java(TM) SE Runtime Environment (build 1.8.0_241-b07)
Java HotSpot(TM) Client VM (build 25.241-b07, mixed mode)

C:\Users\lesli>
```

```
Command Prompt

C:\Users\lesli>javac -version
javac 1.8.0_241

C:\Users\lesli>javac
Usage: javac <options> <source files>
where possible options include:
  -g               Generate all debugging info
  -g:none          Generate no debugging info
  -g:{lines,vars,source}  Generate only some debugging info
  -nowarn          Generate no warnings
  -verbose         Output messages about what the compiler is doing
  -deprecation     Output source locations where deprecated APIs are used
  -classpath <path>  Specify where to find user class files and annotation processors
  -cp <path>        Specify where to find user class files and annotation processors
  -sourcepath <path> Specify where to find input source files
  -bootclasspath <path>  Override location of bootstrap class files
  -extdirs <dirs>      Override location of installed extensions
  -endorseddirs <dirs>  Override location of endorsed standards path
  -proc:{none,only}  Control whether annotation processing and/or compilation is done.
  -processor <class>[,<class>...<class>...] Names of the annotation processors to run; bypasses default discovery process
  -processorpath <path> Specify where to find annotation processors
  -parameters       Generate metadata for reflection on method parameters
  -d <directory>     Specify where to place generated class files
  -s <directory>     Specify where to place generated source files
  -h <directory>     Specify where to place generated native header files
  -implicit:{none,class} Specify whether or not to generate class files for implicitly referenced files
  -encoding <encoding> Specify character encoding used by source files
  -source <release>  Provide source compatibility with specified release
  -target <release>  Generate class files for specific VM version
  -profile <profile> Check that API used is available in the specified profile
  -version          Version information
  -help            Print a synopsis of standard options
  -Akey[=value]    Options to pass to annotation processors
  -X              Print a synopsis of nonstandard options
  -J<flag>         Pass <flag> directly to the runtime system
  -Werror          Terminate compilation if warnings occur
  @<filename>      Read options and filenames from file

C:\Users\lesli>
```

This screenshot shows the GitHub profile page for Leslie Flores. The page includes a profile picture, a bio stating she is a computer science major at Marist College, and a list of pinned repositories: `cmpt220flores` (Java), `CMPT221`, `victoria-cameron/UpInTheGunks` (HTML), and `Operating-Systems` (Python). A contribution graph shows 41 contributions in the last year. The browser address bar shows `github.com/Leslie-Flores`.

This screenshot shows the GitHub repository page for `Leslie-Flores/cmpt440flores`. The page displays the repository name, star/fork counts, and navigation tabs for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The main content area shows the 'Quick setup' instructions for cloning the repository using HTTPS or SSH, and the command-line instructions for creating a new repository or pushing an existing one. The browser address bar shows `github.com/Leslie-Flores/cmpt440flores`.

<https://github.com/Leslie-Flores/cmpt440flores.git>

