Name_____ CSC 210 Java II
Due: 25 November 2025 Fall 2025

Lab 08: Processing Data

Setting Up

1. Download the zip file for lab-08 on Brightspace.

2. Unzip the file to a nice working directory where you usually do your work

3. Enter the unzipped directory. Consider loading it in Visual Studio Code, Eclipse, or the terminal

4. Review the `overview.pdf` to make sure you understand the concepts of population growth using the logistic map.

Exploring the Code Examples

4. **CircleSizeSlider** Compile, execute, and play with the program `CircleSizeSlider.java`. Pay special attention to how the slider is added to the program and what code executes whenever the slider is changed. Make sure you can answer the following questions:

a. What function runs to change the size of the circle?
b. When and how often is the circle redrawn?

5. **Random Programs** Now do the same with the programs `RandomHexPrinter.java` and `RawRandomHexStreamer.java`. When you run `RawRandomHexStreamer`, you will need to press `CTRL-C` to kill the process as it is programmed to run forever.

a. Where are these programs getting their random data from?
b. Is the data they are getting truly random or pseudo random?

6. **LogisticMapGenerator** Use the `LogisticMapGenerator.java` program to generate a csv file with some population data. You must provide a growth rate, an initial population, a number of iterations to compute, and an output file name as arguments at the command line. Remember that the initial population is normalized and so should be a value greater than zero and less than one. You can think of the population size as a value that represents the percentage of the maximum capacity.

Here is an example command:

```
java LogisticMapGenerator 2.5 0.5 50 data.csv
```

Review the output file. If you notice infinities in the output file, it means you may have chosen a non-normalized value for the initial population. Try and generate several csv files in addition to the example above using several different growth rates.

a. What range of growth values lead to extinction?
b. What range of growth values lead to convergence to a constant population?
c. What range of growth values lead to oscillating or chaotic populations?

7. **Visualizing** Compile the programs `CSVtoPNG.java`, `GraphFromCSV.java`, and `GraphAnimationFromCSV.java`. Then execute each program with the data files generated in step 6. Use the visualizations to help you answer the questions a, b, and c in part 6. It is okay to go back and generate additional data if you need to.

a. Make small changes to the three programs, recompile them, and run them to verify that you understand how they are working. Change things like graph colors, plot distances, timer intervals, etc. Use these programs to help you understand how data is read from files and streamed into a program.

8. **GUI Interface** Run the program `RabbitPopulationGraph.java` and verify the conclusions you drew in part 6 by playing with initial conditions and viewing the graph. Note how much faster you can operate within the GUI. Take some time to analyze the code and note the similarities between the smaller component programs in the sketches directory and the code that makes this program work.

a. Make changes to this program and recompile and run it to verify that you understand how it works.
b. What events in the program trigger the graph to be redrawn?
c. Have any of your conclusions needed to be adjusted after playing with the program?

9. **Bifurcation Diagram** Run the program named BifurcationDiagram.java. Shift the maximum and minimum growth values for the program and watch how the diagram changes. This diagram represents possible long term values of a population given particular growth rates. The population value that the system settles on is represented by the y axis. The x-axis represents the range of growth values provided to the logistic map. Areas in the graph that are empty represent values that the population model will not settle on at that particular growth rate.

Notice how the values tend to split periodically as the growth rate increases. This represents that the population will oscillate between those two values over time (or more if there are multiple splits across the vertical axis).

Enjoy exploring this program!

10. **Agentic Simulation** Run the program `RabbitAgentSimulationEmergent.java` . This program attempts to model the logistic map by simulating actual rabbit and carrot objects that interact according to a set of rules instead of just relying on the equation itself.

Take note of the code and how this program is different from the `LogisticMapGenerator`

Review the csv data file that is output by the program and use the visualizer programs to see what is happening. Review the code to see how the rabbit population and the carrots are being manipulated.

Adjust the program's initial values and name the output file something that notes what the initial values were. Do this many times for many notable growth rates in the logistic map. Keep all of the files organized.

a. What is determining the rabbit population limit in this program?
b. How is the carrot population being maintained?
c. Given the data you are receiving output from the simulation with the initial values you've chosen, how close or far is this simulation from the predictions of the logistic map?