

COP2800C Module 7 Graded Programming Assignment – Species Data Class

In this assignment, we will build on our Module 6 work by transitioning from using arrays and methods to grouping related data and functionality into a class. This will reinforce our understanding of objects, classes, and abstraction while improving our program's organization.

NOTE: Use of GitHub is required for this assignment, the starter repository is publicly available for you to fork without using GitHub classroom so there will be no access issues. Instructions will be provided below, you will need to submit your forked GitHub repo URL to the Canvas assignment so I can clone and execute it.

Learning Objectives

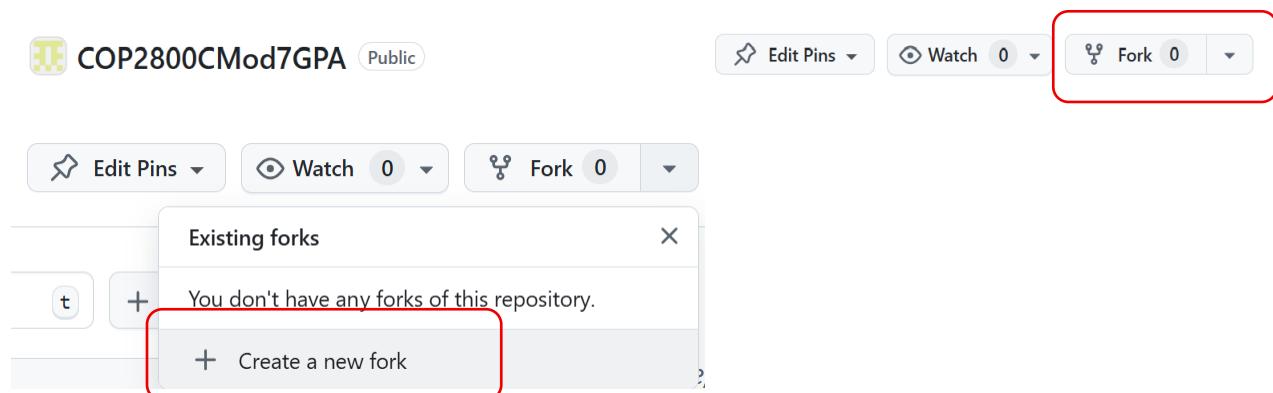
Upon completing this assignment, you will be able to:

- Define and use a class to group related data and operations.
- Implement private fields and public member methods.
- Apply abstraction and information hiding to simplify program design.

Instructions

1. **Fork the GitHub repository.** Forking creates a personal copy of the repository for your use.

- Navigate to <https://github.com/FSCJ-COP2800C/COP2800CMod7GPA>, hit the Fork dropdown and then “Create a New Fork”. Download or clone the forked repository.



- Complete the assignment by following the following instructions (also found in the PDF file in the repo) and submitting your modified files and screen snip to **your** repo (do not submit to mine).
2. **Complete the SpeciesCounter class** (in SpeciesCounter.java)
This class encapsulates the species data and related operations.
Complete the ID header and the tasks shown here:
 - **Private Fields:**

- Find the comment which begins "/* Private fields for encapsulation" and add the following private fields:
- speciesData – A String[] array containing species names.
 - speciesCount – An int[] array representing counts for each species.
- **Public Member Methods:**
- There are 5 public "stub" methods in this file which currently do nothing useful (which is what a stub method is designed to do). **Complete these methods as described below.** Note that the code in these methods is derived from the Module 6 PalmerPenguins.java file; moving code from a single method into multiple smaller methods like this is one form of **refactoring**, which changes the internal structure of code without changing its external behavior.
1. readSpeciesData() – Calls CSVReader.readFile(FILE_NAME, 1) and assigns the result to speciesData. Returns nothing.
 2. initializeSpeciesCount() – Creates a new integer array of size NUM_SPECIES and assigns it to speciesCount. Returns nothing.
 3. isDataEmpty() – Returns true if speciesData is empty, otherwise false.
 4. countSpecies() – Iterates through speciesData and updates speciesCount.
 5. printSpeciesCount() – Prints the count of each species using the predefined constants.

3. Modify main() in PalmerPenguinsM7.java

Complete the ID header and the tasks shown here:

- Find the comment beginning "/* Create an instance of SpeciesCounter" and create an instance of SpeciesCounter named "counter".
- Find the comment beginning "/* Read species data" and call
counter.readSpeciesData();

No other modifications are needed; do not add or modify any of the other code.

4. Test your program

- Verify that your program produces the expected output:

Chinstrap count = 68

Gentoo count = 123

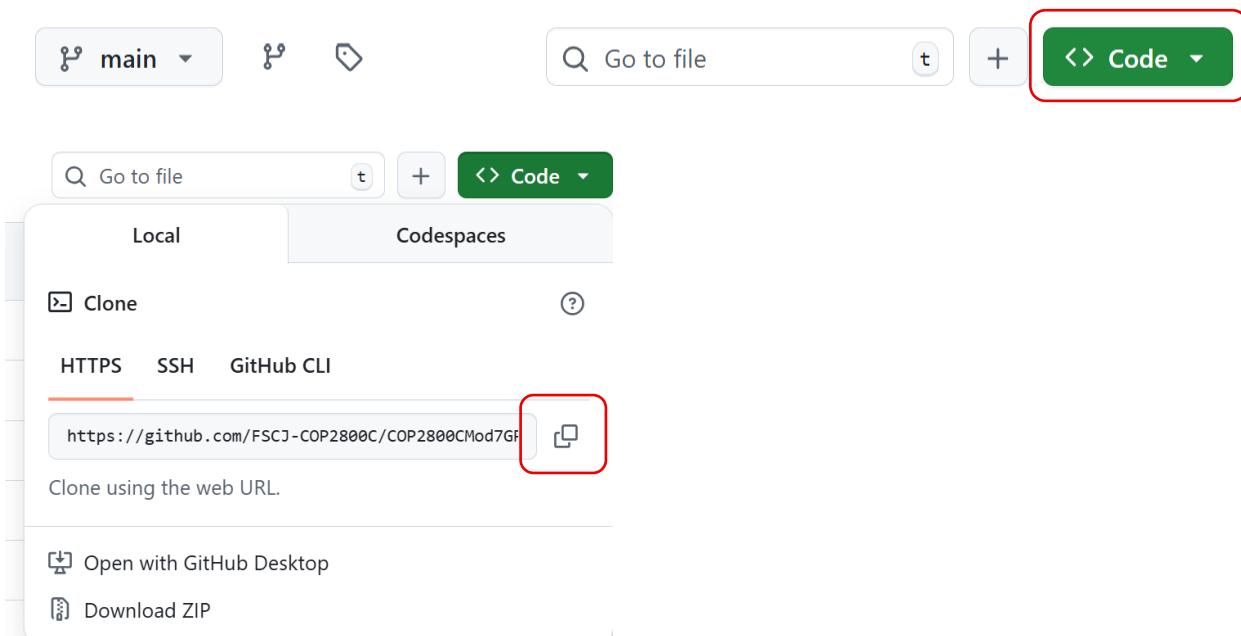
Adelie count = 151

5. Screenshot Your Output

- Take a screenshot of your program's output in the "Run I/O" pane of the jGrasp IDE.
- DO NOT include source code in the screenshot.

6. Submission

- Submit your modified PalmerPenguinsM7.java and SpeciesCounter.java files to **your** GitHub repository, do not submit to mine.
- Do not rename your files or create new folders in the repo; your modified files must overwrite the existing starter files in the repo.
- Upload your output screenshot to the repository.
- Ensure your code follows our class's coding conventions (indentation, comments, line length, meaningful variable names). Be sure your ID headers are completed.
- Add your repository's URL to the Canvas assignment as a comment. Do this by using the Code dropdown in the repo and the copy button as shown here, after you have copied the URL to the clipboard you can paste it into the Canvas assignment using the comment feature.



In the Canvas assignment:

Add Comment