

TEALS Program

[Home](#) | [Curriculum Map](#)

1 Lesson 2.08a — Programming Project Alternative - Lyrics

1.1 Overview

1.1.1 Objectives — *Students will be able to...*

- **Plan and construct** a structured program containing nested loops.

1.1.2 Assessments — *Students will...*

- **Submit** a complete, functional program by the end of next class

1.1.3 Homework — *Students will...*

- **Outline** Chapter 2, omitting BJP 2.5

1.2 Materials & Prep

- **Projector and computer**
- **Student self-help system** (such as C2B4 or student pairing)

Make sure you are set up to grade student notebooks today. If possible, you should only collect 3 – 5 notebooks at a time so students have their notebooks available to reference during programming time.

1.3 Pacing Guide

Section	Total Time
Bell-work and attendance	5min
Introduction & classroom procedures	10min
Programming project	30min
Students trade work, check, & submit	10min

1.4 Procedure

To prepare students for the upcoming unit exam, the next few class periods will be devoted to reinforcing concepts and applying the tools, procedures, and code that were introduced this unit.

1.4.1 Bell-work and Attendance [5 minutes]

1.4.2 Introduction and Classroom Procedures [10 minutes]

1.4.2.1 Classroom Procedures

1. If your computer time requires you to move to another room or to change seating, you should teach and/or review those procedures before introducing the lab material.
2. It's been a few weeks since the last long form programming assignments, so make sure to ask students what the procedures are if they:
 - have gotten stuck (check pseudocode and structure diagram),
 - finished early (move on to challenge questions), or
 - can't remember a coding rule or procedure (check your notes, worksheets, and textbook, C2B4)
3. Unless you have had students submitting work electronically regularly, you should model and review the procedures for submitting electronic work before students begin work.

Introduce the programming project, taking a moment to talk strategy with your class.

PROGRAMMING PROJECT: Write a program that prints out the lyrics of a simple, repetitive song using the minimum number of words in `print/ln()` statements by factoring repetitive words using nested for loops. Here are examples of two songs that can be used: "Row, Row, Row Your Boat", "Jingle Bells"

1.4.2.2 Song 1: "Row, Row, Row Your Boat"

```
Row, row, row your boat
Gently down the stream
Merrily, merrily, merrily, merrily
Life is but a dream
```

```
Row, row, row your boat
Gently down the stream
Merrily, merrily, merrily, merrily
Life is but a dream
```

Song 2: "Jingle Bells"

```
Jingle bells, jingle bells
Jingle all the way
Oh, what fun it is to ride
In a one horse open sleigh
```

```
Jingle bells, jingle bells
Jingle all the way
Oh, what fun it is to ride
In a one horse open sleigh
```

1. In computer science, we look for these patterns as ways to factor out repetition in order to gain efficiency. Students may have other childhood songs from their background that reflects their culture. You can have a discussion on what songs they grew up with and look for patterns in the song. The two songs are just examples. At the teacher's discretion, the students may substitute a song from their childhood. Other songs that may be used are "Happy Birthday" or "You Are My Sunshine".

TIPS: Start with a structure diagram or writing out steps in English as pseudocode. Try to isolate repeated tasks into methods. Include comments in with your code so others can easily understand what the code is supposed to do.

2. Ask your class for suggestions as to how to tackle this programming problem. Students should suggest drawing a structural diagram, building the program one method at a time (iterative development), and following the correction steps on their personal algorithms (debugging).

Procedural decomposition is hard! As a group, ask students to discuss what components go into drawing each line.

- What words repeat?
- What words or phrases repeat on multiple lines?
- What might we want to make its own method that we can call more than once?

1.4.2.3 Emphasize with students

1.4.2.3.1 Big Ideas - Applying the concept of a nested for-loop in drawing artistic patterns

A for-loops allow the repeated execution of one statement (or group of statements). As you iterate through the repetitions, a counter (variable) keeps track of how many times it has already done. The fun thing is: this counter value can also be used for a *nested* for-loop (i.e., have one for-loop inside another for-loop). This combination of using two for-loops means you can write some code to repeat object/patterns that “grow” or “shrink”.

You can think of it this way: A nested loop combination allows you to repeat execution of one statement ... but that one statement *itself* includes a repetition also.

As an alternative to the “Row, Row, Row Your Boat”, “Jingle Bells” song, there are other alternatives that use nested for loops:

- Happy Birthday
- You Are My Sunshine
- There was an Old Lady Who Swallowed a Fly
- Feliz Navidad
- Songs from student's childhood

1.4.3 Programming Project [30 minutes]

The programming project portion has a number of different options.

1.4.3.1 Option 1 Print it as a round

“Row, Row, Row Your Board” is frequently sung in a round. This challenge would be to use the factored song, in this case “Row, Row, Row Your Boat” and print out two sets of lyrics for singing in a round. The following is one round but the project could be extended to generalize to N rounds. Note students do not know how to get input yet so the N would be fixed.

```
1 Row, row, row your boat
2
```

```
1 Gently down the stream
2 Row, row, row your boat
```

```
1 Merrily, merrily, merrily, merrily
2 Gently down the stream
```

```
1 Life is but a dream
2 Merrily, merrily, merrily, merrily
```

1.4.3.2 Option 2 - Alternative Lyrics

1.4.3.2.1 Part 1

Have students can look up the entire lyrics to their song and factor the entire song.

1.4.3.2.2 Part 2

1.4.3.3 Option 3:

Students can make their own song using the same music. Have students think up thier own alternative lyrics and substitute it in the print statements to create their own song. For example, substituting "drive" for "row" and "car" for "boat", the program would now generate "drive, drive, drive your car". Students can pick a popular song they know the chorus to, factor the song in Java, and substutute their own lyrics.

1.4.4 Bonus:

As a bonus to the project, print out the number of words for each line:

```
Row, row, row your boat (5)
Gently down the stream (4)
Merrily, merrily, merrily, merrily (4)
Life is but a dream (5)
```

1.4.5 Students trade work, check, evaluate, and turn in [5 minutes]

At the end of class, have students look over each other's projects before submitting.

Evaluation Question: How many lines of code did this program "save" by using for-loops? How can the code be modified for some other interesting effect, or embellishment?

1.4.5.1 Curricular Competencies - Share and evaluate

It's fun to share artwork. It spurs the imagination. It's valuable to see how others apply the same concept.

Computers are great at doing repetitive tasks. Ask a few questions to evaluate each other's code. For example: How may lines of code did you save by using loops? A well decomposed solution should result in clean, simple code, using as few lines as possible. How can the code be modified for some other interesting effect, or embellishment?

1.5 Accommodation and Differentiation

If you have students who are speeding through this project, you should encourage them to:

- Finish the programming project started in class yesterday.

- Act as student TAs and help struggling classmates (NOTE: you should specifically direct students NOT to give answers, but to help students think of ideas on their own.)

If you have students that are struggling during this class (and you will), resist the urge to help students too much at this stage. Ask leading questions, direct students to their notes, or an example that demonstrates a similar solution, but don't give students the answer here. Resilience/grit is an important emotional tool for solving complex programming problems: the emotional journey students take during these difficult programming problems is as important as the actual coding challenge.

If students are having trouble due to language, pair students up so those with more advanced English can help those that are emergent language learners.

1.6 Forum discussion

[Lesson 2.08 Programming Project \(TEALS Discourse account required\)](#)

formatted by [Markdeep 1.093](#) 