

## Lesson 3.04 — Programming Project

### Overview

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

- **Write** a program that uses parameters, the math class, and returns values.

#### Assessments — *Students will...*

- **Submit** an Equestria program by the end of class.

#### Homework — *Students will...*

- **Read** BJP 3.3 up to “Interactive Programs and Scanner Objects”
- **Complete** Ch.3 self-check questions 18 & 19

### Materials & Prep

- **Projector and computer** (if you are able to/opt to use Eclipse with your students)
- **White paper and markers**
- **Classroom copies** of WS 3.4, Equestria
- **Classroom copies** of Algorithm for Solving Problems
- **Poster or image** of Equestria map with Cartesian coordinates (Poster 3.4)

The handout “Algorithm for Solving Problems” should be used/drilled every time the students are asked to solve a larger, AP Test Section II – type problem. You may find it useful to make a large-format poster to hang in your room so students always see the steps they should use to tackle a large programming problem.

### Pacing Guide

Section	Total Time
Bell-work and attendance	5min
Introduction to programming project	5min
Student programming time	45min

### Procedure

Over the next few weeks, we’ll be introducing students to larger programming projects to meet the AP Computer Science A lab requirements. It is in your and your students’ best interest if you encourage them to help themselves and each other before seeing you. Depending on the length of your class periods, this lab may take 2 class periods to complete.

#### Bell-work and Attendance [5 minutes]

#### Introduction to Programming Project [10 minutes]

Begin with an introduction to today’s programming project.

- Students should complete this programming project individually. As a whole group, review the Algorithm for Solving Problems sheet. Read the sheet out loud, and briefly model the steps as you review the programming assignment. Read the programming assignment to the class, taking time to pause between each of the three requirements outlined in the question.
- Ask students what their very first steps should be according to the problem-solving algorithm.
- Require that pseudocode/comment documentation be submitted. Include these as part of the lab grade.
- Remind students to tackle one part of the problem at a time. It is OK if they leave pseudocode in while they solve a different part of the problem, and they don’t have to solve all of parts of the program in any particular

order.

- They should refer to their notes, textbooks, and posters/work around the room to help them come up with ideas/solutions to programming problems.
- Sometimes it's a good idea to step away from a problem for a few minutes if they're stuck—work on a different method and return to where they were stuck later.

## Student Programming Time [45 minutes]

In an email, on the projector, or as a handout (WS 3.4), give students the following questions to work on individually, or in pairs.

---

### Emphasize with students...

**Content - Problem decomposition - Management of complexity** By the end of this activity you will have created quite a complex program. But notice how the activity takes you through the smaller steps in building the program?

The finished program is broken up into smaller steps, then they're each completed individually. When they're put together they create the larger program. This type of decomposition allows the programmer to manage complexity. Rather than considering the whole problem and program, the programmer can focus on each small component, then assemble them together at the end.

Writing programs with methods allows us to divide the program up into smaller parts quite easily.

---

---

## PROGRAMMING PROJECT

**Exercise 1** Princess Luna and Celestia are going on a tour of the kingdom to greet the other citizens of Equestria. Their tour takes them on a circular path (shown on map). Write a method called `roadTrip` that (1) accepts as a parameter the diameter of the circular path, and (2) returns the length of the trip.

The equation for circumference is:  $C = d$

Java has a math constant called `Math.PI`.

**Exercise 2** Write a method called `distance` that (1) accepts four integer coordinates: `x1`, `y1`, `x2`, `y2` as parameters, (2) computes the distance between points (`x1`, `y1`) and (`x2`, `y2`) on the map, and (3) returns that distance.

The equation for the distance is:  $\text{sqrt}((x2 - x1)^2 + (y2 - y1)^2)$ .

Test out your program by writing a main method that calls the `distance` method for each of the following pairs of cities. Your main method should output the value returned by the `distance` method.

1. Distance from Baltimore to Manehattan =
2. Distance from Los Pegasus to Neighagra Falls =
3. Distance from the Badlands to Ponyville =

**Exercise 3** Write a program that helps Princess Luna plan a 3-stop tour of Equestria. Choose any 3 locations in Equestria, as defined by their `x` and `y` coordinates on the map. Your program should output the distance between the three destinations.

You should use the `distance` methods you wrote for Exercise 2.

**Exercise 4** Write a method called `totalTrip` that accepts parameters for 3 locations (each containing coordinates) and returns the total distance traveled by visiting all 3 locations and returning to the starting location. You should use the distance methods you wrote in Exercise 2 and you can choose any 3 locations in Equestria.

Extra credit: make this program compute 4 locations instead!

---

## Accommodation and Differentiation

For students who complete the lab early, ask them to flesh out their program by adding a method that will calculate the sum of the distances travelled between three cities in Exercise 2.

If you suspect that students will struggle with procedural decomposition, have them work in groups to figure out a course of action before beginning. Procedural decomposition is one of the hardest skills for students to acquire, so it is critical that you allow students to fight through the process.

If additional scaffolding is needed, you might list all of the parts of the program, or have students come up with the parts (“figure out how to write the equation in Java using the math class,” “create some parameters to pass to a return method,” “write the framework for a method to return the distance value”) out of order, then give the students some time to organize the steps themselves. As a whole group, you can then come to consensus on what steps need to be approached, and what order components should appear in the final program.

## Forum discussion

Lesson 3.04 Programming Project (TEALS Discourse account required)