Coding and Geometry

Student Version

Authors:

Michelle Hills

Fernando Alegre

Juana Moreno

Summer 2015

LA-SiGMA

Louisiana Alliance for Simulation-Guided Materials Applications

LSU

Coding and Geometry: Student Introduction

What?

Coding and Geometry is a method of illustrating the use of coding in the basics of high school geometry. You will manipulate programs by learning different skills taught step by step throughout the lessons. From basic skills such as drawing points on a display to more advanced functionality such as calculating the perimeter of an arbitrary triangle, these lessons encompass the goal of incorporating computer science skills into the high school mathematics classroom.

How to access Haskell Files:

Windows:

1. Create a folder in Documents named Haskell
2. Download the files from the website.
3. Open the zip file.
4. Click Extract All Files.
    a. Browse
    b. Save it under Documents/Haskell
5. Extract

Mac:

1. Create a folder in Documents named Haskell
2. Download the files from the website.
3. Drag the files from downloads into the Haskell folder.
4. Then double click to unzip the file.

As you manipulate and create your own files (exercises), you will want to save them in a folder (YourName_Coding) in the Haskell/k12math-student folder on the computer. At the end of the lesson if the computer is not your own, you will want to copy your folder onto your USB drive.

When you come back to manipulate and create files again in another lesson, just copy the folder from your USB to the Haskell/k12math-student folder on the computer. Repeat this process when using a school or someone else's computer.

Getting started with the terminal:

After installing the three programs listed above, open the terminal and perform the following commands:

Teacher & Student

1. Type cd Documents/Haskell/k12math-student
2. Type util/prepare

Running Code:

Windows:

1. Open GitBash
2. cd Documents/Haskell/k12math-_____ (the blank should either be student or teacher)
3. ls
    a. If running a program already installed in k12math-_____, type rungeo prog/_____ /lesson1a.hs (the blank should either be student or teacher)
    b. If running a program you manipulated and saved in your personal folder, type
    rungeo yourname_coding/lesson1a.hs

MAC:

1. Open the terminal (applications/utilities)
2. Open GitBash
3. cd Documents/Haskell/k12math-_____ (the blank should either be student or teacher)
4. ls
    a. If running a program already installed in k12math-_____, type ./rungeo prog/_____ /lesson1a.hs (the blank should either be student or teacher)
    b. If running a program you manipulated and saved in your personal folder, type
    ./rungeo yourname_coding/lesson1a.hs

Name: _____ Date: _____ Class: _____

Lesson 1: Points, Lines, Segments

*All exercises should be opened in the editor and ran in the terminal as stated in the introduction.

Lesson 1 Part 1: Open the editor and then open lesson1a.hs. Look at the code and write down what you understand in the code. Run the code.

Notes:

Questions:

1. What function draws the points?

2. What function draws the labels?

Exercise: *Draw and Label three additional points in the program.*

Save the program as yourname_lesson1b.hs

Run the program to check.

Questions:

How do you think we would draw a segment between points A and B?

Lesson 1 Part 2: Open the editor and then open lesson1c.hs.

Exercise: *Determine what the program will draw.*

Run the program to check.

Questions:

What function do you think we would use to draw lines?

Lesson 1 Part 3: Open the editor and then open lesson1d.hs

Exercise: *Determine what the program will draw.*

Run the program to check.

Lesson 1 Ending Exercises:

Exercise: *Open lesson1e.hs*

Sketch a drawing of what you think is happening in this program.

Run the program to check your answer.

Exercise: *Open lesson1f.hs.*

Complete the program to draw the following objects:

1. *Segment AB*
2. *Line BC*
3. *Segment CA*
4. *Message needs to read "Segments and Lines"*

Save the program as yourname_lesson1bc.hs.

Run the Program to check.

Exercise: *Using four random points, how many segments would you need to connect each point with the other three points on the graph?*

Manipulate one of the programs above to check your answer.

Save the program as yourname_lesson1g.hs

Run the program to check.

Exercise: *Repeat the above exercise for 5 random points.*

Save the program as yourname_lesson1h.hs

Run the program to check.

Name: _____ Date: _____ Class: _____

Lesson 2: Intersections

*All exercises should be opened in the editor and ran in the terminal as stated in the introduction.

Lesson 2 Part 1: Open the editor and then open lesson2a.hs Look at the code and write down what you understand in the code. Run the code.

Notes:

Questions:

1. What does the function `find_apart` do?

2. What does the `drop 3 points` do?

Lesson 2 Part 2: Open the editor and the open lesson2b.hs. Look at the code and then run the program in the terminal

Questions:

3. What does the program do?

4. What does `p` represent?

5. What does the function `line_line (a,b) (c,d)` used for?

6. How many points could the function `line_line (a,b) (c,d)` return?

Lesson 2 Part 3: Open lesson2c.hs. Look at the code carefully**.**

Questions:

1. What do you think `intersects = line_circle (c,d) (a,b)` will compute?

   Run the program and compare your answers to these questions.

2. What do the letters correspond to in `drawCircle (a,b)`?

3. What does (c,d) correspond to in `line_circle (c,d) (a,b)`?

Lesson 2 Part 4: Open lesson2d.hs.

Exercise: *Determine what the program will draw.*

Run the program to check.

Lesson 2 Ending Exercises:

Exercise: *Open lesson2e.hs. Remove the word* `undefined` *to get started. Create a program to draw the following:*

1. *Circle with center a and point b on the circle.*
2. *Line AC*
3. *Label the intersection points of the circle and line as D and E.*

Save the program as yourname_lesson2e.hs

Run the program to check.

Name: _____ Date: _____ Class: _____

Lesson 3: Midpoint & Distance

*All exercises should be opened in the editor and ran in the terminal as stated in the introduction.

Lesson 3 Part 1: Open the editor and then open lesson3a.hs. Look at the code and write down what you understand in the code. Run the code.

Notes:

Questions:

1. What does the program do?

2. Why did the programmer use `a'` in the language?

Exercise: *Draw and Label the two missing midpoints of the other segments. Label appropriately as discussed in the question above.*

Save your program as yourname_lesson3b.hs

Run the program to check.

Exercise: *Manipulate the program you just created, lesson3b.hs, to draw the 3 segments connecting the midpoints.*

Save your program as yourname_lesson3c.hs

Run the program to check.

Lesson 3 Part 2: Open the editor and then open lesson3d.hs.

Question:

This program should look familiar to lesson3a.hs.

1. What do you think the program does differently than lesson3a.hs?

   Run the program.

2. What does the function `dist b c` do?

3. What does the function `message` do?

Exercise: *Open the editor and then open lesson3e.hs.*

1. *What does the program do?*

   Run the program to check.

2. *What does the function* `messages` *do?*

Lesson 3 Ending Exercises:

Exercise: *Open lesson3d.hs*

Manipulate the program to do the following:

1. *Draw all three midpoints of the segments.*
2. *Draw all of the segments connecting the midpoints.*
3. *Show the measure the distances of all 3 original sides.*
4. *Show the measure the distances of the 3 segments connecting the midpoints.*

Save the program as yourname_lesson3f.hs

Run the program to check

Name: _____ Date: _____ Class: _____

Lesson 4: Angles

*All exercises should be opened in the editor and ran in the terminal as stated in the introduction.

Lesson 4: Open the editor and then open lesson4a.hs. Look at the code and write down what you understand in the code. Run the code.

Notes:

Questions:

1. What does the program do?

1. What does the b stand for in `shownum (angle a b c)`?

Exercise: *Open lesson4b.hs in the editor. Look at the program and sketch a picture of what the program is doing?*

Run the program to check.

Questions:

1. What do you notice about the two angles that are drawn?

2. What does the program draw?

3. How does the program calculate where `a'` should be?

4. What does the function `drawArc (a,o,a')` do?

Lesson 4 Ending Exercises:

Exercise: *Open yourname_lesson3f.hs*

Manipulate the program to show the angle measures of the 3 angles in the large triangle and the three angles in the triangle created by the midpoints.

Save the program as yourname_lesson4c.hs

Run the program to check

Question: *What do you notice about the angle measures in the large triangle compared with the smaller triangle?*

Name: _____ Date: _____ Class: _____

Lesson 5: Perpendicular & Parallel Lines

*All exercises should be opened in the editor and ran in the terminal as stated in the introduction.

Lesson 5 Part 1: **Notes:**

Exercise: *Open lesson5a.hs in the editor. Create a sketch of what the program is drawing.*

Run the program to check your answer.

Exercise: *Manipulate lesson5a.hs to label the intersection point as* `E` *and display the message "Perpendicular Lines."*

Save the program as yourname_lesson5b.hs

Run the program to check your answer.

Lesson 5 Part 2:

Question:

How can we draw a line parallel to line AB in the previous exercises?

Exercise: *Manipulate yourname_lesson5b.hs to draw line FG parallel to line AB and display the message "Parallel Lines."*

Save the program as yourname_lesson5c.hs

Run the program to check your answer.

Lesson 5 Ending Exercises:

Exercise: *Manipulate yourname_lesson5c.hs to draw another line HI parallel to line AB and display the message "Parallel Lines." Your display should have three parallel lines and one perpendicular line.*

Save the program as yourname_lesson5d.hs

Run the program to check your answer.

Exercise: *Manipulate yourname_lesson5d.hs to remove the perpendicular line and labels of the perpendicular line from the display.*

Save the program as yourname_lesson5e.hs

Run the program to check your answer.

Name: _____ Date: _____ Class: _____

Lesson 6: Perimeter & Area

*All exercises should be opened in the editor and ran in the terminal as stated in the introduction.

Lesson 6: **Notes:**

Questions:

What function(s) could we use to help us create a program to calculate the perimeter of a figure?

Exercise: *Using lesson1f.hs in the editor, create a program to draw a triangle using three points. The program should also show the lengths of the three sides and show the perimeter of the triangle.*

Save the program as yourname_lesson6a.hs

Run the program to check.

Exercise: *Manipulate the above program for a four sided figure and only display the perimeter as a message.*

Note: Instead of using `take 4 points`*, use the function* `quadrilateral points`*.*

Save the program as yourname_lesson6b.hs

Run the program to check.

Lesson 6 Ending Exercises:

Exercise: *Open lesson6c.hs to calculate the area of a triangle. Show the area as a message.*

Save the program as yourname_lesson6c.hs

Run the program to check.

Name: _____ Date: _____ Class: _____

Lesson 7: Equilateral Triangle

*All exercises should be opened in the editor and ran in the terminal as stated in the introduction.

Lesson 7: Open the editor and open lesson7a.hs. Look at the code carefully.

Notes:

Exercise: *Determine what the program will draw.*

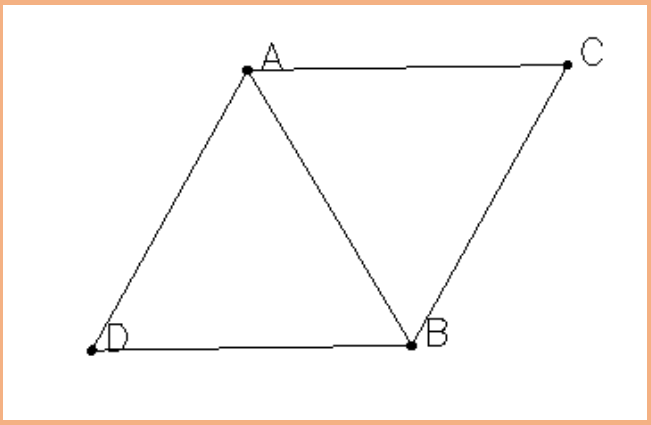
Run the program to check.

Questions:

1. What shape did the program create?
2. How is point c computed in the program?
3. Why is the list `[c,d]` assigned to `circle_circle (a,b) (b,a)` ?
4. Why is `d` not shown when you run the program?

Lesson 7 Ending Exercises:

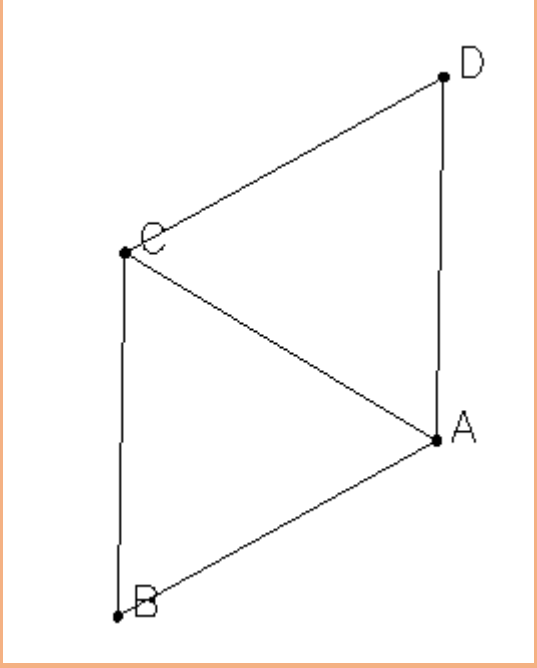
Exercise: *Manipulate the program lesson7a.hs to draw a picture with the characteristics of the one below:*



Save the program as yourname_lesson7b.hs

Run the program to check.

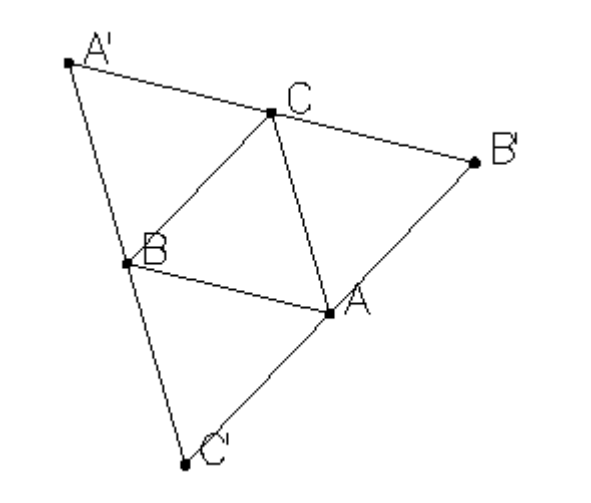
Exercise: *Manipulate the program lesson7b.hs to draw a picture with the characteristics of the one below:*



Save the program as yourname_lesson7c.hs

Run the program to check.

Exercise: *Manipulate the program lesson7a.hs to draw a picture with the characteristics of the one below:*



Save the program as yourname_lesson7d.hs

Run the program to check.

Name: _____ Date: _____ Class: _____

Lesson 8: Transformations

*All exercises should be opened in the editor and ran in the terminal as stated in the introduction.

Lesson 8 : Open the editor and then open lesson8a.hs. Look at the code and write down what you understand in the code. Run the code.

Notes:

Questions:

1. What function makes the points move?

2. Why in the translate function is `a'` written as `(x1+x,y1+y)`?

Exercise: *Manipulate the program lesson8a.hs to draw segment between the translated points.*

Save the program as yourname_lesson8b.hs

Run the program to check.

Exercise: *Manipulate the program lesson8b.hs to translate a quadrilateral 2 units right and 5 units up. The quadrilateral should be made of points (0,0), (3,0), (3,3),(0,3).*

Save the program as yourname_lesson8c.hs

Run the program to check.

Lesson 8 Part 2: Open the editor and then

open lesson8d.hs

Exercise: *Determine what the program will draw.*

Run the program to check.

Question:

1. Why is `a' = (-y,x)`?

2. What would happen if you rotate (9,8) 180 degrees counter-clockwise?

Exercise: *Manipulate the program lesson8d.hs that will rotate a quadrilateral 90, 180, or 270 degrees. The program should have all three functions at the bottom. Write this specific program to rotate the quadrilateral (1,1), (1,4), (4,4), (4,1) 180 degrees counter-clockwise.*

Save the program as yourname_lesson8e.hs

Run the program to check.

Lesson 8 Part 3: Open the editor and then open lesson8f.hs.

Questions:

1. Why is the function mirrors used as a transformation of reflection?

2. Why is `a' = (x, -y)`?

Exercise: *Manipulate the program lesson8f.hs to reflect triangle (1,2),(4,2),(3,3) over the x-axis.*

Save the program as yourname_lesson8g.hs

Run the program to check your answer.

Exercise: *Manipulate the program lesson8g.hs that could reflect a quadrilateral over the x or y axis. Both mirrors functions should be listed at the bottom of the program. For this specific program reflect the quadrilateral (1,2), (1,4), (3,4), (3,2) over the y-axis.*

Save the program as yourname_lesson8h.hs

Run the program to check your answer.

Lesson 8 Ending Exercises:

Exercise:

Question: How can we draw a random triangle in a program?

Manipulate yourname_lesson8g.hs to reflect a random triangle over the x-axis.

Save the program as yourname_lesson8i.hs.

Run the Program to check your answer.

Exercise:

Question: How do we perform a dilation on a figure?

Manipulate yourname_lesson8g.hs to dilate the triangle by a scale factor of 2 rather than reflect the triangle.

Save the program as yourname_lesson8j.hs

Run the program to check your answer.

Name: _____ Date: _____ Class: _____

Coding Project

Create a program using 10 functions with at least one function from each lesson we have covered. Refer to your handouts, notes, programs, and reference sheet for help. You should open file project.hs in Notepad++ to help you get started. Before you begin, remove the word `undefined` from the program.

You will be graded based on required elements, content-accuracy, organization, and creativity/originality. Consult the rubric for more information.

Save your project as yourname_project.hs