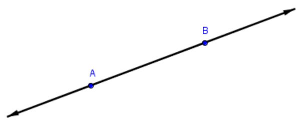
# Drawing the Line



A class is a software bundle of variables (fields) and methods. A class is also a blueprint for objects, which are actual instances of the class (they contain all the variables and the methods defined in the class, and the variables will have actual values). Bundling variables and methods into a class creates a custom type of thing!

As always, more information is available on our website.

First! Create a package called `line`.

1. Make a new class called Point that will be a bundle of a couple variables and methods, for defining a single point (an X and Y location, like a point on a graph).

    a. Instance variables:

    - `private double x` – the point's X coordinate
    - `private double y` – the point's Y coordinate

    b. Methods

    - `public Point(double xVal, double yVal)` – constructor, initializes the instance variables to the values of the parameters and creates a new Point object.
    - Add getter and setter methods for BOTH variables.
    - `public String toString()` – method that returns a printable String, for printing the state of a point. Copy/paste the following code inside your method:

    ```
    return "{x: " + x + ", y: " + y + "}";
    ```

2. Create a class called Line. This class models a line (a line is simply a connection between two points). A Line is comprised of two Points, along with a couple methods.

    a. Instance variables

    - `private Point firstPoint` – first point in the line
    - `private Point secondPoint` – second point in the line

    b. Methods

    - `public Line(double x1, double y1, double x2, double y2)` – constructor, initializes the two Point instance variables to new Point objects, given the values of the parameters and creates a new Line object.
    - `public Line(Point p1, Point p2)` – (overloaded) constructor, initializes the two Point instance variables to the value of the parameters and creates a Line object.
    - `public double getSlope()` – returns the slope of this line.
    - Add getter and setter methods for BOTH variables. The getter methods should return the point's toString() method.

- `public String toString()` – method that returns a String that allows a Line to be printed in a useful way. Copy/paste the following code inside your method:

```
return "<Point one: " + firstPoint.toString() + ", Point two: " + secondPoint.toString() + ">";
```

3. Create a new class called Runner with a `public static void main(String[] args)` method and complete the following exercises:

   a. Make a Point object called `p1`, with 23.14 and 4.87 supplied (for `x` and `y`) to the constructor.

   b. Make a Point object called `p2`, with 15.2 and 6.87 supplied (for `x` and `y`) to the constructor.

   c. Make a Line object called `line1`, with `p1` and `p2` supplied to the constructor.

   d. Make a Line object called `line2`, with the values 12.45, 8.1, 9.2, and 14.7 supplied to the constructor.

   e. Print the `line1` object (print a call of `line1.toString()`).

   f. Print a call of the `line2` object's `toString()` method.

   g. Print a call to the `getSlope()` method on the `line1` object.

   h. Print a call to the `getSlope()` method on the `line2` object.

   i. Call the Runner class' `main()` method; your output should look like the following. Fix any errors you have.

```
<Point one: {x: 23.14, y: 4.87}, Point two: {x: 15.2, y: 6.87}>
<Point one: {x: 12.45, y: 8.1}, Point two: {x: 9.2, y: 14.7}>

line1 slope >>> -0.251889168765743
line2 slope >>> -2.0307692307692307
```

## Comparing two Line objects (optional)

Add a method `public boolean isEqual(Line other)` to the Line class that returns true if the Line object the method is called on is the same as the supplied `other` Line (has the same Points). Begin by adding a similar method in the Point class that will compare two Points for equivalency. Examples:

```
        Line line1 = new Line(new Point(1, 1), new Point(1, 1));
        Line line2 = new Line(new Point(2, 2), new Point(2, 2));
        Line line3 = new Line(new Point(2, 2), new Point(2, 2));

        line1.isEqual(line2) >>> false
        line2.isEqual(line3) >>> true

/*
The isEqual() methods are instance methods - they must be called on an object.  In the Point
class, this method will compare the Point object calling the method ("this" Point) to the
Point object other, supplied as a parameter.

        Point first = new Point(5, 5);
        Point second = new Point(10, 10);
        first.isEqual(second); //would return false
*/
```