Review:
Disturbances: Outside influences that act on a robot while it tries to stay on a path

Trajectory: The path the robot is supposed to be following. It is made up of waypoints that have position, and maybe velocity, or time, or other variables associated with each of them.

Proportional Feedback Control: It is a common solution for keeping a robot on a trajectory. This is when you compare the current value to the desired value to determine the error. The output signal is proportional to the size of the error.

We'll use the **vector format** notation for a line and define two in particular:

$$\overrightarrow{line}_{start\ to\ end} = \overrightarrow{line}_{S\rightarrow E} = \begin{bmatrix} End_x - Start_x \\ End_y - Start_y \end{bmatrix}$$

$$\overrightarrow{line}_{start\ to\ robot} = \overrightarrow{line}_{S\rightarrow R} = \begin{bmatrix} robot_x - start_x \\ robot_y - start_y \end{bmatrix}$$

We can use the **cross product** to determine which side we are on:

$$side = \overrightarrow{line}_{S\rightarrow E} \times \overrightarrow{line}_{S\rightarrow R} = \begin{bmatrix} 0 \\ 10 \end{bmatrix} \times \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

$$\begin{bmatrix} A \\ C \end{bmatrix} \times \begin{bmatrix} B \\ D \end{bmatrix} \rightarrow AD - BC$$

Negative means we are on the right side, positive on the left side.
We can find our distance from the line using the dot product and the normal vector:

$$distance = \frac{\left|\overrightarrow{line}_{S\rightarrow R} \cdot \overrightarrow{normal}\right|}{\left\|\overrightarrow{normal}\right\|}$$

We can find the normal by taking the original line, swapping the x and y coordinates, and negating one of them.

$$\overrightarrow{Line} = \begin{bmatrix} X \\ Y \end{bmatrix} \rightarrow \overrightarrow{Normal} = \begin{bmatrix} -Y \\ X \end{bmatrix}$$

For the magnitude, we take the **Pythagorean theorem** to find the total length:

$$a^2 + b^2 = c^2$$

Where a and b are the x and y components, and c is the total length.

$$\begin{bmatrix} A \\ C \end{bmatrix} \cdot \begin{bmatrix} B \\ D \end{bmatrix} \rightarrow AB + CD$$

For the **dot product**, we multiply the x terms of both arguments, multiply the two y terms, then add the results.

Final Step:
Translate the error signal into a steering angle. We use a constant factor called the proportional gain, to both scale and translate the error into the correct units.

Error × proportional gain = command
E × K = command

K = proportional gain
E = Error

$$K = \frac{\_\_\_\_ ^\circ}{\text{meter}}$$

Abstraction in Computer Science is the idea of limiting out some information that is not necessary for the user to see and only showing the user the critical information. In a way, abstraction is “information hiding.”

Name: ______________________________ Class: _________________

Challenge Questions:

If our robot had a remote controller, what information would be hidden from the user and what information would the user have access to?
The user would be able to access the controls, up down, left right, ect. The user would not have access to the math calculations that the robot was doing or the communication of directions between different parts of the robot.
Why is it more important to hide information in increasingly complex robotic systems?
The more complex the system, the more abstraction is needed because there is more information to sift through. There are often layers of abstraction where information is grouped into smaller and smaller sections of information that can be seen by the user on different levels.

Name: ______________________________ Class: __________________