Milestone 3: Make the spaceship move in a straight line and turn.

For this milestone, you will make your spaceship move forward at a constant speed, and turn right and left in response to keys pressed by the user.

Reminder: your spaceship should have 0 rotation and be facing towards the east (the right side of the screen) when the game begins.

Decide which keys the user should press to have the spaceship move forward, and turn left and right. If you user keys other than the arrow keys, put this in the comment when you submit your code, so that I know how to test it.

Create a method `public void move()` in your `Ship` class. This class will change the position of your ship if the forward key is being held, and change the rotation of your ship if either of the turn keys are being held.

To keep track of which key is currently being pressed, add boolean instance variables to your `Ship` class for the forward, and left and right turn keys. Each variable will be set to true if the corresponding key is being pressed, and to false if it's not.

Add code to your `move()` method to make your `Ship` move forward when the forward key variable is true. To do this, think about what variable holds your ship's current position? How do you access the x coordinate and y coordinate of its current position?

To test out your code so far, set the forward key variable to true, and the other two to false. Call `move()` in `Asteroids.paint(..)`. (To figure out how to call move, think about whether it is a static or non-static method) Your ship should move forward and disappear off the edge of the screen.

Once you have your ship moving, change your code so that once the ship moves off one side of the screen, it wraps around and appears on the other side of the screen. i.e. if the ship moves off the right edge of the screen, it should re-enter on the left side; if it moves off the top of the screen, it should re-enter on the bottom, etc. Test your code again for the current case (ship is only moving forward).

As discussed in class (in the class on interfaces), we will make `Ship` implement the `KeyListener` interface (https://docs.oracle.com/javase/7/docs/api/java/awt/event/KeyListener.html) so that it can respond to keys on the keyboard being pressed. To do this, we need to change the header of the `Ship` class appropriately, and add three methods: `public void KeyPressed(...)`, `public void KeyReleased(...)`, and `public void KeyTyped(...)`. Even though we will leave `KeyTyped()` empty, it must be added to satisfy the conditions of the `KeyListener` interface.

All of the `KeyListener` methods take a `KeyEvent` object (https://docs.oracle.com/javase/7/docs/api/java/awt/event/KeyEvent.html) as their parameter. This object contains information about which key was pressed or released. You can get the key code, which is a number representing the key pressed or released by calling the non-static method `getKeyCode()` for the `KeyEvent` instance that was passed in. The list of the constants corresponding to the key codes is on the `KeyEvent` API page (https://docs.oracle.com/javase/7/docs/api/java/awt/event/KeyEvent.html). Fill in the `KeyPressed()` and `KeyReleased()` methods so that when the forward key is pressed or released, its corresponding boolean variable changes appropriately. See the class `Test`, which we did in class, for an example.

The `Canvas` object, which is the superclass of `Game` and hence `Asteroids`, generates `KeyEvents` when a key is pressed, released, or typed. We hence need to register our new `KeyListener` with `Asteroids`, so that it will know where to send these event objects. We do this by adding the code `this.addKeyListener(nameOfShipInstance);` in the constructor of `Asteroids`.

Initialize the boolean variable for the forward key being pressed to false. Test your code to make sure that your ship moves when the forward key is pressed.

Add code in `move()` to make the `Ship` rotate if a turn key is being held. Test that the ship is rotating (even if it is not moving in the correct direction). Also test that if the ship moves off the screen in any direction, it re-enters from the other side.

Now we will get your ship moving in the direction it is facing:

If your ship doesn't move in the correct direction even before pressing a turn key, then you need to change the direction that your ship is facing when it starts.

If your ship moves in the correct direction when you start, but then keeps moving in that same direction even after you press a turn key, then you need to use some trigonometry make sure you are incrementing the x and y coordinates the correct amount to go in the desired direction. Specifically, if you are currently incrementing the x and y coordinates by the same amount, you need to instead multiple the increase in the x coordinate by `Math.cos(Math.toRadians(rotation))` and multiply the increase in the y coordinate by `Math.sin(Math.toRadians(rotation))`.