- You must have **at least** these methods and instance variables. You must submit comprehensive, **modular test code** and **proper style, modular class methods** electronically by the deadline on Canvas.
- `public class Cannonball` A class that simulates a cannonball firing in the air.

Fields

Modifier and Type	Field and Description
private static String	CANNON_BALL
private static double	CLOCK_TICK_MILLISEC
private static double	DEFAULT_X_POSITION
private double	elapsed
private static String	EMPTY_SPACE
private static double	GRAVITY_METERSperSECONDsquared
private static String	GROUND_OFFSET
private static double	MAX_ELAPSED_SECONDS
private static int	SCALE
private double	xPos
private double	xVel
private double	yPos
private double	yVel

- ## Constructor Summary

Constructor and Description
Cannonball() Initializes a cannonball to DEFAULT_X_POSITION.
Cannonball(double xPos) Initializes a cannonball to the given x position.

- ## Method Summary

Modifier / Type	Method and Description
private static void	displayGraphic(double yPos) Display "text graphic" of current position above "ground"
private static void	displayLocation(double elapsed, double xPos, double yPos) Display current elapsed time and x and y coordinates
double	getX() Get the x position of the cannonball.
double	getY() Get the y position of the cannonball.
void	moveBall(double sec) Moves the cannonball for the given time unit based on its x and y velocities, also updates y velocity for the effect of gravity.
private static String	repeat(String str, int times) Repeat string **str times** copies.
void	shoot(double launchAngle, double initialVelocity) Simulates a cannonball being fired given the initial angle and velocity.

Example, Sample runs:

```
Enter a starting angle, 1-89 degrees: 45
    Enter a firing velocity (0-20 meters/second): 10
    sec: 0.10, x: 0.71, y: 0.71     | *
    sec: 0.20, x: 1.41, y: 1.32     |  *
    sec: 0.30, x: 2.12, y: 1.83     |  *
    sec: 0.40, x: 2.83, y: 2.24     |   *
    sec: 0.50, x: 3.54, y: 2.55     |   *
    sec: 0.60, x: 4.24, y: 2.77     |   *
    sec: 0.70, x: 4.95, y: 2.89     |   *
    sec: 0.80, x: 5.66, y: 2.91     |   *
    sec: 0.90, x: 6.36, y: 2.83     |   *
    sec: 1.00, x: 7.07, y: 2.66     |   *
    sec: 1.10, x: 7.78, y: 2.38     |   *
    sec: 1.20, x: 8.49, y: 2.01     |   *
    sec: 1.30, x: 9.19, y: 1.54     |  *
    sec: 1.40, x: 9.90, y: 0.97     | *
    sec: 1.50, x:10.61, y: 0.31     | *
    sec: 1.60, x:11.31, y:-0.46     | *


 Enter a starting angle, 1-89 degrees: 45
    Enter a firing velocity (0-20 meters/second): 20
    sec: 0.10, x: 1.41, y: 1.41     |  *
    sec: 0.20, x: 2.83, y: 2.73     |   *
    sec: 0.30, x: 4.24, y: 3.95     |    *
    sec: 0.40, x: 5.66, y: 5.07     |      *
    sec: 0.50, x: 7.07, y: 6.09     |       *
    sec: 0.60, x: 8.49, y: 7.01     |        *
    sec: 0.70, x: 9.90, y: 7.84     |        *
    sec: 0.80, x:11.31, y: 8.57     |         *
    sec: 0.90, x:12.73, y: 9.20     |          *
    sec: 1.00, x:14.14, y: 9.73     |          *
    sec: 1.10, x:15.56, y:10.16     |           *
    sec: 1.20, x:16.97, y:10.50     |           *
    sec: 1.30, x:18.38, y:10.73     |           *
    sec: 1.40, x:19.80, y:10.87     |           *
    sec: 1.50, x:21.21, y:10.91     |           *
    sec: 1.60, x:22.63, y:10.86     |           *
    sec: 1.70, x:24.04, y:10.70     |           *
    sec: 1.80, x:25.46, y:10.45     |           *
    sec: 1.90, x:26.87, y:10.09     |           *
    sec: 2.00, x:28.28, y: 9.65     |          *
```

display ends after a default of 2 seconds of flight.