

PMR: The program was easy in my opinion. It was good review of other programs. Also the instructions for this assignment made it easy to create this program.

/\*\*

\* Project title: 09.04 Challenge Program

\*

\* Purpose of Project: Calculates the trajectory of a projectile based on launch angles and launch velocities.

\*

\*

\* @version 12/12/2019

\*

\* @author Anika Jallipalli

\*/

public class Catapult {

private double velocity, degrees, distance;

public Catapult(double v, double deg) {

velocity = v;

degrees = deg;

}

public double getVelocity(){

return velocity;

}

public double getDegrees(){

return degrees;

}

public double getDistance(){

return distance;

}

public void calcDistance(){

distance = (Math.pow(velocity, 2) \* Math.sin( 2 \* Math.toRadians(degrees)) / 9.8);

}

}

***TESTER***

/\*\*

\* Project title: 09.04 Challenge Program

\*

\* Purpose of Project: Calculates the trajectory of a projectile based on launch angles and launch velocities.

\*

\* @version 12/12/2019

\*

\* @author Anika Jallipalli

\*/

public class CatapultTester {

public static void main(String[] args) {

Catapult[][] catapults = new Catapult[7][7];

int speed;

for(int row = 0; row < catapults.length; row++) {

speed = 20 + 5 \* row;

for(int column = 0; column <catapults.length; column++)

{

catapults[row][column] = new Catapult(speed, 25 + 5 \* column);

}

}

System.out.println(" Projectile Distance (feet) ");

System.out.println("MPH 25 Deg 30 Deg 35 Deg 40 Deg 45 Deg 50 Deg 55 Deg ");

System.out.println();

System.out.println("======================================================================");

for (Catapult[] catapult : catapults) {

System.out.printf("%2.0f ", catapult[0].getVelocity());

for (int column = 0; column < catapults.length; column++) {

catapult[column].calcDistance();

System.out.printf("%8.1f ", catapult[column].getDistance());

}

System.out.printf("\n");

}

}

}