

AKDENİZ ÜNİVERSİTİ 2023 FALL PHYSİCS FOR COMPUTER SCİENCE I FİNAL EXAM HOMEWORK PART

1) The question 2 a was answered collaboratively with my friend Burak Yalçın (20220808069) using ChatGPT.

Here are the conversations with ChatGPT.

You

>>>Here are the codes that answer the question 2b that I wrote before.<<<
Selam. Eğik atılan bir merminin uçuş süresini, maximum yüksekliğini ve menzilini hesaplayan kodlar yazdım. Şimdi merminin alacağı yörüngeyi çizmemiz gerekiyor. Freechart kütüphanesiyle böyle bir şey yapabilir miyiz? Kısaca kod çalışınca bir panel açılacak. Daha sonra panelin içinde öğrenci numarası girebileceğimiz bir textbox olacak. Öğrenci numarasını oraya girip Draw butonuna bastıktan sonra panel üzerinde merminin yörüngesi çizilecek. Ek olarak daha önceden hesapladığımız uçuş süresini, maximum yüksekliği ve menzilini de panelin içinde virgülden sonra 2 basamak olacak şekilde yazdırsın.



ChatGPT

Tabii ki, aşağıda, javada yörüngeleri çizmek için JFreeChart kütüphanesini kullanan bir örnek kod bulabilirsin. Ancak, bu örnek kod sadece bir başlangıçtır ve kendi projenize entegre etmeden önce ihtiyaçlarınıza uygun olarak özelleştirmeniz gerekebilir.

```
java Copy code  
  
import org.jfree.chart.ChartFactory;  
import org.jfree.chart.ChartPanel;  
import org.jfree.chart.JFreeChart;  
import org.jfree.chart.plot.XYPlot;  
import org.jfree.chart.renderer.xy.XYLineAndShapeRenderer;  
import org.jfree.data.xy.XYSeries;  
import org.jfree.data.xy.XYSeriesCollection;  
  
import javax.swing.*;  
  
public class TrajectoryChart extends JFrame {
```

-- THE CODE CONTINUES --

All of the project is available on my [GitHub](#).

And I used ChatGPT to determine the number of digits after the comma.
Ex. 28,4689 instead of 28,468985463254

You

```
private void updateLabels(String studentNum) {
    initialVelocityLabel.setText("Initial Velocity: " + Final.initialVelocity(studentNum));
    timeOfFlightLabel.setText("Time of Flight: " +
    Final.calculateTimeOfFlight(studentNum));
    maxHeightLabel.setText("Max Height: " + Final.calculateMaxHeight(studentNum));
    rangeLabel.setText("Range: " + Final.calculateRange(studentNum));
}
```

bunları nasıl virgülden sonra 2 basamak gelecek şekilde yapabilirim



ChatGPT

Java'nın `String.format` veya `DecimalFormat` gibi araçları kullanarak sayıları belirli bir formatta biçimlendirebilirsiniz. İşte örneğin `DecimalFormat` kullanarak `updateLabels` metodunu güncelleyebilirsiniz:

java

Copy code

```
import java.text.DecimalFormat;

private void updateLabels(String studentNum) {
    DecimalFormat decimalFormat = new DecimalFormat("#.##");

    double initialVelocity = Final.initialVelocity(studentNum);
    double timeOfFlight = Final.calculateTimeOfFlight(studentNum);
    double maxHeight = Final.calculateMaxHeight(studentNum);
    double range = Final.calculateRange(studentNum);

    initialVelocityLabel.setText("Initial Velocity: " + decimalFormat.format(initialVelocity));
    timeOfFlightLabel.setText("Time of Flight: " + decimalFormat.format(timeOfFlight));
    maxHeightLabel.setText("Max Height: " + decimalFormat.format(maxHeight));
    rangeLabel.setText("Range: " + decimalFormat.format(range));
}
```

Bu örnekte, `DecimalFormat("#.##")` ile ondalık sayıları iki basamaklı bir formatta biçimlendiriyoruz. `decimalFormat.format(initialVelocity)` gibi bir çağrı, ondalık sayıyı belirtilen formatta bir dizeye dönüştürür. Bu şekilde, etiketlerinizdeki sayılar virgülden sonra iki basamaklı olacaktır.

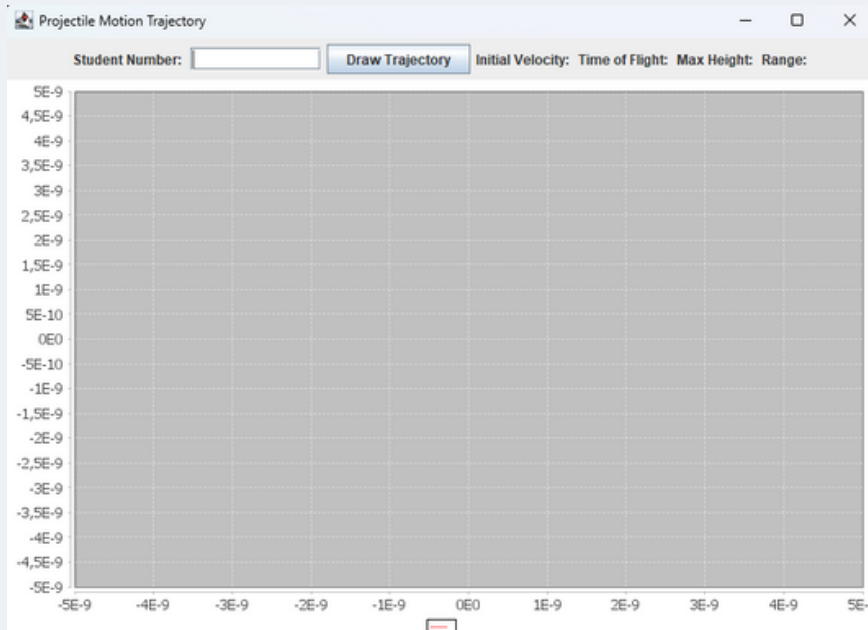
Additionally, I used ChatGPT to assist me in installing the FreeChart library. I also watch videos.

2)

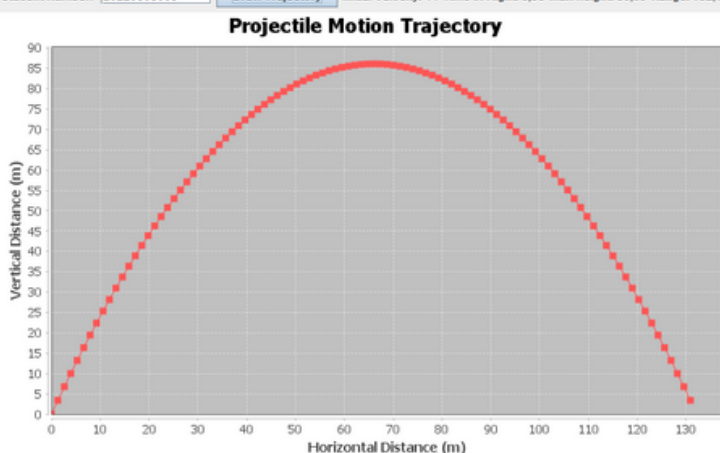
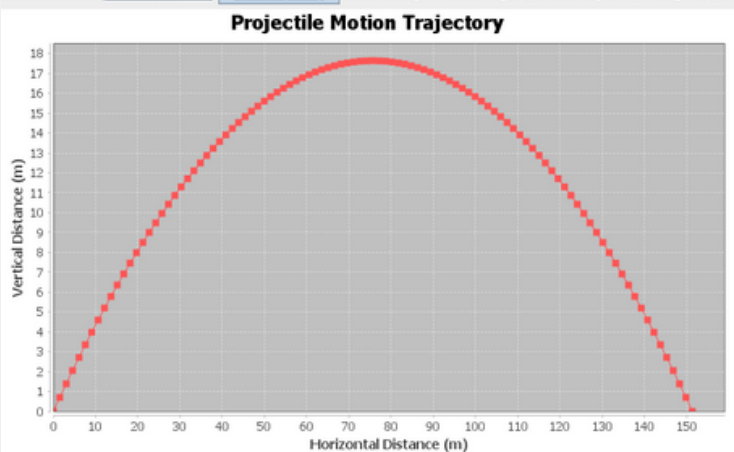
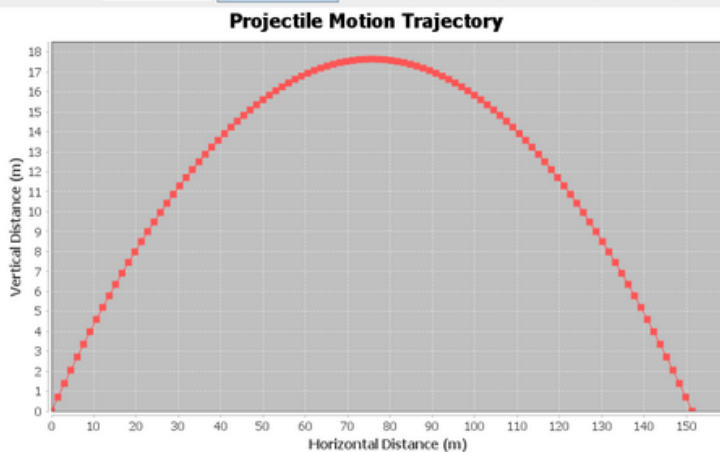
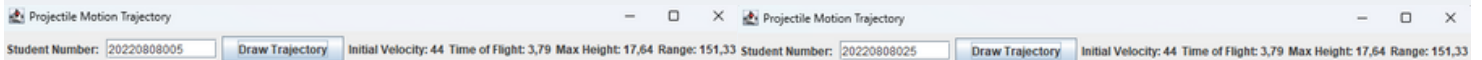
a) Plotting the trajectory of the projectile motion.
The result when I run the program:

TrajectoryChart.java

And enter the student number to calculate and plot.



The angle with horizontal axis must be the same as added from itself if last 2 number is lower than 20.



Different angle and velocity values must provide different motion. It looks not different at first. But the scaling on the graph changes depending on the entered number.

2)

b) This is Final.java file in src folder and it calculates initial velocity, angle with horizontal axis and time of flight using student number as a parameter. The code below:

```
public class Final {

    final public static double g = 9.8;

    public static int initialVelocity(String studentNum){
        String initialVelocityString = studentNum.substring(2, 4);
        int initialVelocity = 2*Integer.parseInt(initialVelocityString);
        return initialVelocity;
    }

    public static int angleWithHorizontal(String studentNum){
        int length = studentNum.length();
        String angleString= studentNum.substring(length-2,length);
        int angle = Integer.parseInt(angleString);

        if(angle<20)return angle+20;
        else return angle;
    }

    public static double calculateTimeOfFlight(String studentNum ){

        double x = Math.toRadians(angleWithHorizontal(studentNum));
        double sin = Math.sin(x);
        return (2*initialVelocity(studentNum)*sin)/g;
    }

    public static double calculateMaxHeight(String studentNum ){
        double x = Math.toRadians(angleWithHorizontal(studentNum));
        double sin = Math.sin(x);

        double t =calculateTimeOfFlight(studentNum)/2; // half of flight time

        return (initialVelocity(studentNum)*sin*t)-((g*t*t)/2);
    }

    public static double calculateRange( String studentNum){
        double x = Math.toRadians(angleWithHorizontal(studentNum));
        double cos = Math.cos(x);

        double t =calculateTimeOfFlight(studentNum);

        return initialVelocity(studentNum)*cos*t;
    }

    public static void main(String[] args) {

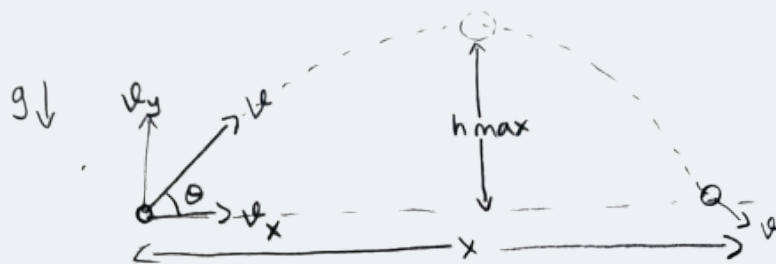
    }

}

} //class Final
```

2)

b) I did this calculations to use them in Final.java file.



Time of flight

$$V_f = V_0 + at$$

$$0 = v_y - gt \Rightarrow t = \frac{v_y}{g}$$

t is half of flight

$$2t \rightarrow \frac{2v_y}{g}$$

$$v_y = v \cdot \sin \theta$$

so time of flight:

$$\frac{2 \cdot v \cdot \sin \theta}{g}$$

Max height

$$\Delta x = V_0 t + \frac{1}{2} at^2$$

$$h = v_y t - \frac{1}{2} gt^2$$

$$h = \frac{v \cdot \sin \theta \cdot t}{2} - \frac{\frac{1}{2} gt^2}{4}$$

Axis Y

Range

$$\Delta x = V_0 t + \frac{1}{2} at^2$$

$$X = v_x \cdot t$$

$$x = v \cdot \cos \theta \cdot t$$

Axis X

for the graph (x, h)

3) The program will calculate and print the velocity just before impact when throwing an object downward using energy formulas. The input values are height, mass, gravitational acceleration and initial velocity. I did this calculations to use my Java program.



$$U = m \cdot g \cdot h \Rightarrow mgh$$

$$K = \frac{1}{2} m v_i^2 \Rightarrow \frac{1}{2} m v_i^2$$

$$\text{Total} = mgh + \frac{1}{2} m v_i^2$$

$$U = 0$$

$$K = \frac{1}{2} m v_f^2$$

$$\text{Total} = \frac{1}{2} m v_f^2$$

$$mgh + \frac{1}{2} m v_i^2 = \frac{1}{2} m v_f^2$$

$$gh + \frac{1}{2} v_i^2 = \frac{1}{2} v_f^2$$

$$v_f^2 = 2gh + v_i^2$$

$$v_f = \sqrt{2gh + v_i^2}$$

3) This is CalculateVelocity.java file. The code below:

```
public class CalculateVelocity {

static final double g = 9.8;

static double calculateFinalVelocity(double h , double initialVelocityOnYAxis,double mass){

    double vi= initialVelocityOnYAxis;
    double vFinalSquare = (2*g*h)+(vi*vi);
    double vFinal = Math.sqrt(vFinalSquare);
    return vFinal;
}

public static void main(String[] args) {
System.out.println("Enter the height");

Scanner input = new Scanner(System.in);
System.out.println("Welcome to the final velocity calculation program.");
System.out.print("You will be launching an object downward, ");
System.out.println("and we will determine its final velocity just before impact.");
System.out.println("-----");
System.out.println("");

System.out.print("Please enter the mass of the object: ");
double mass = input.nextDouble();
System.out.print("Enter the velocity with which you are launching the object: ");
double velocity = input.nextDouble();
System.out.print("Enter the height from which you are launching the object: ");
double height = input.nextDouble();
System.out.println("");
System.out.println("Calculating...");
System.out.println("");
double vv= calculateFinalVelocity(height, velocity, mass);
System.out.printf("The final velocity just before impact is %.4f%n", vv);

input.close();

}
```

The output is in the next page.

CalculateVelocity.java

Welcome to the final velocity calculation program.
You will be launching an object downward, and we will determine its final velocity just before impact.

Please enter the mass of the object: 25
Enter the velocity with which you are launching the object: 07
Enter the height from which you are launching the object: 100

Calculating...

The final velocity just before impact is 44,8219

Welcome to the final velocity calculation program.
You will be launching an object downward, and we will determine its final velocity just before impact.

Please enter the mass of the object: 1895
Enter the velocity with which you are launching the object: 07
Enter the height from which you are launching the object: 100

Calculating...

The final velocity just before impact is 44,8219

As can be seen, I also wanted to show that it is independent of mass.

It is done here. As I said the all files is in [My GitHub Repositories](#). The freeChart library is also in lib folder. All code can be run directly (The 2a with freeChart Library) with downloading it (I hope). Additionally, thank you for the whole semester. The classes I attended were very instructive and good. I don't think I paid enough attention yet. I believe I will do this next semester.

CSE-111-Physics-I / src /

efekurucay Update 7 hours ago

Name	Last commit message	Last commit date
..		
CalculateVelocity.java	Update	7 hours ago
Final.java	Update	7 hours ago
TrajectoryChart.java	Update	7 hours ago