# 

# 

# 

# Course: ENSF 614 - Fall 2023

# Lab B01: Lab 1

# Instructor: Mahmood Moussavi

# Student Name: Braden Tink

# Submission Date: September , 2023

## 

## 

## 

## 

## 

## 

## 

## Lab1 Exe A (Omited)

## Lab1 Exe B

#### Code

/\*

\* lab1exe\_B.cpp

\* ENSF 614 Lab 1, exercise B

\* Created by Mahmood Moussavi

\* Completed by: Braden Tink

\* Submission Date: Sept 20, 2023

\*/

#include <iostream>

#include <cmath>

using namespace std;

const double G = 9.8; /\* gravitation acceleration 9.8 m/s^2 \*/

const double PI = 3.141592654;

void create\_table(double v);

double Projectile\_travel\_time(double a, double v);

double Projectile\_travel\_distance(double a, double v);

double degree\_to\_radian(double d);

void create\_table(double v){

double angle;

double radians;

double distance;

double time;

cout << "Angle t d\n";

cout << "(deg) (sec) (m)\n";

angle = 0;

while ( angle <= 90){

radians = degree\_to\_radian(angle);

distance = Projectile\_travel\_distance(radians, v);

time = Projectile\_travel\_time(radians, v);

cout << angle << " " << time << " " << distance << "\n";

angle += 5.0;

}

}

double Projectile\_travel\_time(double a, double v){

return ((2\*v\*sin(a))/G);

}

double Projectile\_travel\_distance(double a, double v){

return ((v\*v)/G)\*sin(2\*a);

}

double degree\_to\_radian(double d){

return (d\*PI)/180;

}

int main(void)

{

double velocity;

cout << "Please enter the velocity at which the projectile is launched (m/sec): ";

cin >> velocity;

if(!cin) // means if cin failed to read

{

cout << "Invlid input. Bye...\n";

exit(1);

}

while (velocity < 0 )

{

cout << "\nplease enter a positive number for velocity: ";

cin >> velocity;

if(!cin)

{

cout << "Invlid input. Bye...";

exit(1);

}

}

create\_table(velocity);

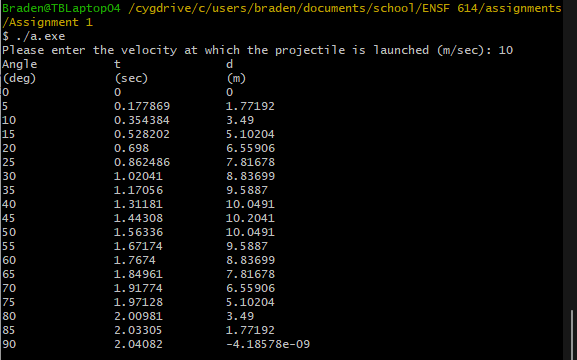
return 0;

}

#### 

#### 

#### Output



## Lab1 Exe C

(Omitted)

## Lab1 Exe D

### Part 1 (Omitted)

### Part 2

## 

## Lab1 Exe E

#### Code

/\*

\* lab1exe\_E.cpp

\* ENSF 614 Lab 1, exercise E

\* Created by Mahmood Moussavi

\* Completed by: Braden Tink

\* Submission Date: Sept 20, 2023

\*/

#include <iostream>

using namespace std;

void time\_convert(int ms\_time, int \*minutes\_ptr, double \*seconds\_ptr);

/\*

\* Converts time in milliseconds to time in minutes and seconds.

\* For example, converts 123400 ms to 2 minutes and 3.4 seconds.

\* REQUIRES:

\* ms\_time >= 0.

\* minutes\_ptr and seconds\_ptr point to variables.

\* PROMISES:

\* 0 <= \*seconds\_ptr & \*seconds\_ptr < 60.0

\* \*minutes\_ptr minutes + \*seconds\_ptr seconds is equivalent to

\* ms\_time ms.

\*/

int main(void)

{

int millisec;

int minutes;

double seconds;

cout << "Enter a time interval as an integer number of milliseconds: ";

// printf("Enter a time interval as an integer number of milliseconds: ");

cin >> millisec;

if (!cin) {

cout << "Unable to convert your input to an int.\n";

exit(1);

}

cout << "Doing conversion for input of " << millisec <<" milliseconds ... \n,";

time\_convert(millisec, &minutes, &seconds);

/\* MAKE A CALL TO time\_convert HERE. \*/

cout << "That is equivalent to " << minutes << " minute(s) and " << seconds << " second(s).\n";

return 0;

}

/\* PUT YOUR FUNCTION DEFINITION FOR time\_convert HERE. \*/

void time\_convert(int ms\_time, int \*minutes\_ptr, double \*seconds\_ptr){

int remainder;

if (ms\_time <= 60000){

\*minutes\_ptr = 0;

\*seconds\_ptr = (ms\_time / 1000);

}

else{

remainder = ms\_time % 60000;

\*minutes\_ptr = (ms\_time /60000);

\*seconds\_ptr = (remainder / 1000);

}

}

#### Output

