**Course: ENSF614 – Fall 2023**

**Lab #:**

**Instructor: M. Moussavi**

**Student Name : Sieu Eric Diep**

**Submission Date : Sep 18, 2023**

**Exercise B – Source Code:**

/\*

\* lab1exe\_B.cpp

\* ENSF 614 Lab 1, exercise B

\* Completed by: Sieu Diep

\* Date: Sep 13, 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);

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;

}

void create\_table(double v){

double distance = 0, time = 0, radian = 0;

printf("%6s \t %6s \t %s \n","Angle","t","d");

printf("%6s \t %6s \t %s \n","(deg)","sec","m");

for(int angle = 0; angle <= 90; angle += 5){

radian = degree\_to\_radian(angle);

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

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

printf("%6d \t %6.3f \t %.3f \n", angle, time, distance);

}

}

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

double time = 2\*v\*sin(a)/G;

return time;

}

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

double distance = v\*v/G \* sin(2\*a);

return distance;

}

double degree\_to\_radian(double d){

double radian = d \* PI/180;

return radian;

}

**Exercise B Program output:**

**A screenshot of a computer

Description automatically generated**

**Note: the last row has a -0 value due to the lack of precision of the float, i.e. it is off by a extremely small amount that is very close to 0.**

**Exercise D2:**

**A white paper with writing on it

Description automatically generated**

**Exercise E - Source Code:**

/\*

\* lab1exe\_E.cpp

\* ENSF 614 Lab 1 Exercise E1

\* Student name: Sieu Eric Diep

\* Date: Sep 13, 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", millisec;

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

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

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)

{

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

\*seconds\_ptr = (double)(ms\_time % 60000) / 1000;

}

**Output:**

**A screen shot of a computer

Description automatically generated**