**Course: ENSF 614** - Fall 2023  
**Lab #:** Lab 1  
**Instructor:** Mahmood Moussavi  
**Student Name:** Christian Valdez  
**Submission Date:** September 20, 2023

**Exercise B:**

/\*

\* Filename: lab1exe\_B.cpp

\* Assignment: Lab 1 rxercise B

\* Completed by: Christian Valdez

\* Submission date: Sep 20, 2023

\*/

#include <iostream>

#include <iomanip>

#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);

double angleDeg;

double angleRad;

int main(void) {

double velocity;

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

cin >> velocity;

if (!cin) {

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) {

angleDeg = 0;

double t;

double d;

cout << "Angle (deg)" << "\t t (sec)" << "\t d (m)" << endl;

cout << fixed << setprecision(6);

while (angleDeg < 95) {

angleRad = degree\_to\_radian(angleDeg);

t = Projectile\_travel\_time(G, v);

d = Projectile\_travel\_distance(G, v);

cout << angleDeg << "\t " << t << "\t " << d << endl;

angleDeg += 5;

}

}

double degree\_to\_radian(double d) {

return angleDeg \* (PI / 180);

}

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

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

}

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

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

}

**A screenshot of a computer

Description automatically generated**

**Exercise D2**

A diagram of a bar code

Description automatically generated

**Exercise E**

/\*

\* Filename: lab1exe\_E.cpp

\* Assignment: Lab 1 rxercise E

\* Completed by: Christian Valdez

\* Submission date: Sep 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: ";

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;

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

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

return 0;

}

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

if (ms\_time >= 0) {

\*minutes\_ptr = ms\_time / (1000 \* 60);

\*seconds\_ptr = (double)(ms\_time % (1000 \* 60)) / 1000;

}

}

A screenshot of a computer

Description automatically generated