/\* This program calculates the various parameters of Projectile Motion

Input: Initial velocity or Range

Output: Maximum height reached by projectile and time it takes to strike the target

\*/

#include <stdio.h>

#include <iostream.h>

#include <math.h>

#include <stdlib.h>

#include <string.h>

#include <conio.h>

#include <dos.h>

#include <graphics.h>

#include <fstream.h>

#define EXIT\_PROGRAM 5

#define ACC\_DUE\_TO\_GRAVITY 9.8

#define PI 3.414

#define OPTIMAL\_ANGLE 45

const char\* fName = "PROJ.DAT"; //file to store data

class projectile{

private:

int choice;

float velocity;

float angle;

float range;

float maxRange;

float timeOfFlight;

int bestAngle;

float height;

float mHeight;

fstream fileObj;

public:

projectile() {

choice = 0;

velocity = 0;

angle = 0;

range = 0;

maxRange = 0;

timeOfFlight = 0;

bestAngle = 0;

mHeight = 0;

height = 0;

}

~projectile(){

}

void getParameters(int choice);

float convertToRadians();

void calcMaxRange();

void calcRange();

void maxHeight();

void calcHeight();

void timeToHitTarget();

void calcInitialVelocity();

int deleteDataFromFile();

void displayResults(int choice);

float getVelocity(){

return velocity;

}

float getAngle(){

return angle;

}

float getHeight(){

return height;

}

float getFlightTime(){

return timeOfFlight;

}

float getRange(){

return range;

}

}; // end of class projectile

int printBannerAndMenu(void);

int storeValuesInFile(projectile&);

int displayDataFromFile();

int deleteDataFromFile();

void main(){

int choice = 0;

clrscr();

for(; ;) {

clrscr();

char resp = 'A';

projectile myProj;

choice = printBannerAndMenu();

switch(choice){

case 1:

myProj.getParameters(choice);

myProj.calcRange();

myProj.calcMaxRange();

myProj.calcHeight();

myProj.timeToHitTarget();

myProj.displayResults(choice);

cout << endl << endl << "Do you want to save these values. Press Y to save or any key to return to Menu: ";

cin >> resp;

if(resp == 'y' || resp == 'Y'){

if(storeValuesInFile(myProj) == -1){

cout <<" Failed to save values in file" << endl;

break;

}

cout << endl << "Values have been successfully saved" << endl;

}

else

cout << "Values Not Saved" << endl;

break;

case 2:

cout << endl << endl;

clrscr();

myProj.getParameters(choice);

myProj.calcInitialVelocity();

myProj.calcHeight();

myProj.timeToHitTarget();

myProj.displayResults(choice);

cout << endl << endl << "Do you want to save the values. Press Y to Save or any key to return to Menu: ";

cin >> resp;

if(resp == 'y' || resp == 'Y'){

if(storeValuesInFile(myProj) == -1){

cout <<" Failed to save values in file" << endl;

break;

}

cout << endl << "Values have been successfully saved" << endl;

}

else

cout << "Values Not Saved" << endl;

break;

case 3:

displayDataFromFile();

break;

case 4:

if(deleteDataFromFile() == -1){

cout << "Could not delete values from File" << endl;

}

else

cout << endl << endl << "Successfully deleted values from file" << endl;

break;

case 5:

cout << "\n\n\n\nThanks for using our system. Bye Bye !!" << endl;

break;

default:

cout << "Invalid choice entered. Please enter a valid choice" << endl;

break;

}//end of switch

if (choice == EXIT\_PROGRAM)

break;

sleep(6);

} //end of for

}

// User Defined functions

float projectile::convertToRadians(){

return ((angle/180.0)\*(PI));

}

void projectile::calcRange(){

range = ((velocity\*velocity)\* sin(2 \* convertToRadians()))/ACC\_DUE\_TO\_GRAVITY;

return;

}

void projectile::calcMaxRange(){

maxRange = ((velocity\*velocity))/ACC\_DUE\_TO\_GRAVITY;

return;

}

void projectile::calcHeight(){

float ang = 0;

ang = sin(convertToRadians());

height = (velocity \* velocity \* ang \* ang)/(2 \* ACC\_DUE\_TO\_GRAVITY);

return;

}

void projectile::timeToHitTarget(){

float ang = 0;

ang = sin(convertToRadians());

timeOfFlight = (2 \* velocity \* ang)/ACC\_DUE\_TO\_GRAVITY;

return;

}

int printBannerAndMenu(){

int choice = 0;

textcolor(GREEN);

cout << "\n\n\n\t";

cprintf("Welcome to the Class 12 Computer Science Project on Projectile Motion");

cout << "\n\n\n\nPress 1 to enter projectile velocity and angle" << endl;

cout << "\tThis will display the Range, Maximum height and time to flight" << endl;

cout << "\n\nPress 2 to enter Range" << endl;

cout << "\tThis will display the required projectile initial velocity and angle" << endl;

cout << "\n\nPress 3 to display stored values" << endl;

cout <<"\tThis will display previously stored values" << endl;

cout << "\n\nPress 4 to delete stored values" << endl;

cout <<"\tThis will delete previously stored values" << endl;

cout << "\n\nPress 5 to Exit" << endl;

cin >> choice;

return choice;

}

void projectile::getParameters(int choice){

clrscr();

cout << "\n\n\n\t";

cprintf("Welcome to the Class 12 Computer Science Project on Projectile Motion");

switch(choice){

case 1: //if choice == 1;

cout << "\n\n\n\nEnter Initial Velocity of Projectile in m/s: ";

cin >> velocity;

cout << "\n\nEnter the launch angle in degrees: ";

cin >> angle;

break;

case 2:

cout << "\n\n\n\nEnter the Range of the Projectile in meters:" ;

cin >> range;

angle = OPTIMAL\_ANGLE;

break;

default:

cout <<"Wrong choice" << endl;

}

return;

}

void projectile::calcInitialVelocity(){

velocity = sqrt(ACC\_DUE\_TO\_GRAVITY\*range);

return;

}

int storeValuesInFile(projectile& proj){

fstream fileObj(fName, ios::out| ios::app);

if(!fileObj){

cout <<"File Open failed" << endl;

return -1;

}

fileObj.write((char \*)&proj, sizeof(proj));

fileObj.flush();

fileObj.close();

return 0;

}

void projectile::displayResults(int choice){

clrscr();

cout << "\n\n\n\t";

cprintf("Welcome to the Class 12 Computer Science Project on Projectile Motion");

cout.precision(2);

cout << "\n\n" << "Velocity is:" << velocity << " m/s and angle is:"

<< angle << " deg"<< endl;

cout<<"\n\n\n" << "Range is:" << range << "m" << endl;

cout <<"Height reached by projectile: " << height

<< " m" << endl;

cout << "Time to hit the Target: " << timeOfFlight

<< "secs" << endl;

if(choice == 1){

cout<< "\n\nMaximum range at this velocity is when:" << endl;

cout << "\t" << "Angle = 45 degrees" <<endl;

cout << "\t" << "Range = " << maxRange << endl;

}

return;

}

int displayDataFromFile(){

ifstream fileObj(fName, ios::binary);

projectile obj;

clrscr();

if(!fileObj){

cout <<"Cannot open file" << endl;

return -1;

}

cout << "\n\n\n\t";

cprintf("Welcome to the Class 12 Computer Science Project on Projectile Motion");

cout << endl << endl;

cout << "\n\n\t\tv(m/s)\tA(rad)\tR(m)\tT(sec)\tH(m)" << endl;

fileObj.read((char \*) &obj, sizeof(obj));

while(fileObj){

cout.precision(2);

cout << "\t\t" << obj.getVelocity() << "\t"<< obj.getAngle()

<<"\t" << obj.getRange() <<"\t"<< obj.getFlightTime()

<<"\t"<< obj.getHeight() << endl;

fileObj.read((char \*) &obj, sizeof(obj));

}

fileObj.close();

return 0;

}

int deleteDataFromFile(){

fstream fileObj;

projectile obj;

clrscr();

fileObj.open(fName, ios::out); // default ios:: out would truncate file

if(!fileObj){

cout <<"Cannot open file" << endl;

return -1;

}

fileObj.close();

cout << "\n\n\n\t";

cprintf("Welcome to the Class 12 Computer Science Project on Projectile Motion");

return 0;

}