**CECS 282**

**Week 4 - Lab 1**

**Taylor Girard**

**Evan Haut**

**Kenry Yu**

Code:

**main**

//Authors: Taylor Girard, Evan Haut, Kenry Yu

//Contributions:

// Evan Haut: Main

// Taylor Girard: setAltitude(), crash()

// Kenry Yu: I set up the header file, both default and overload constructor, and the display function.

#include <iostream>

#include "Airplane.h"

#include <time.h>

using namespace std;

int main() {

srand(time(NULL));

double numCrashed = 0;

Airplane plane1 = Airplane("B17", 600, 500, 2000);

Airplane plane2 = Airplane("MH340", 700, 200, 1500);

for(int i = 0; i < 1000; i++)

{

if(plane1.crash(plane2))

{

plane1.display();

plane2.display();

cout << endl;

numCrashed++;

}

plane1.setAltitude();

plane2.setAltitude();

}

double percent = numCrashed / 10;

cout << "Crashed a total of " << numCrashed << " times, or " << percent << "%.";

}

**Airplane.h**

#ifndef AIRPLANE\_H

#define AIRPLANE\_H

#include <string>

class Airplane{

private:

std::string model;

int altitude;

int minAltitude;

int maxAltitude;

public:

Airplane();

Airplane(std::string name, int height, int minAl, int maxAl);

void display();

void setAltitude();

bool crash(Airplane plane);

int getAltitude();

};

#endif

**Airplane.cpp**

#include "Airplane.h"

#include <iostream>

using namespace std;

Airplane::Airplane(){

this->model = "N/A";

this->altitude = 0;

this->minAltitude = 0;

this->maxAltitude = 0;

};

Airplane::Airplane(string name, int height, int minAl, int maxAl){

this->model = name;

this->altitude = height;

this->minAltitude = minAl;

this->maxAltitude = maxAl;

};

//Displays the model and altitude

void Airplane::display(){

cout << this->model <<" flying at "<< this->altitude<<" feet.\n";

};

//Sets the altitude to a random value between the max and min altitude

void Airplane::setAltitude(){

int temp\_altitude = rand() % (this->maxAltitude - this->minAltitude) + this->minAltitude;

this->altitude = temp\_altitude;

};

//Returns the planes altitude

int Airplane::getAltitude(){

return altitude;

};

bool Airplane::crash(Airplane plane){

if (abs(this->altitude - plane.getAltitude()) < 200){

return true;

}

return false;

}

Outputs:



