**CMPSC 201 Project Report**

**Project 1**

**974343924 – Ibrahim Ibrahim, Youssef El Minawi, Diego Ibarra**

**09/28/2018**

**Pseudocode**

Write a program that will take the following inputs from the user:

* The magnitude of the velocity with which the Red Bird was flung, in m/s
* The angle with respect to the horizontal of the velocity vector, in degrees
* The horizontal distance to the structure you're trying to knock down with the Red Bird, in m
* The height of the structure you're trying to knock down with the Red Bird, in m

You may assume the following:

* Assume the height from which the Red Bird is flung is a constant of 5 m. Make sure the user knows this information to scale the height of the structure accordingly.
* Your toss will be perfectly along the line between you and your target, i.e. you don't need to worry about the toss going to the left or the right.
* Since it is a video game, there's no wind. There is no air resistance/wind resistance that we must take into account, i.e. the "ignore friction and air resistance" clause that's standard in most introductory physics books applies here.

Your objective is to compute and output:

1. The time it takes for the Red Bird to reach the target.
2. The magnitude and direction of the velocity when the Red Bird reaches the target distance.
3. The height of the Red Bird when it reaches the target distance
4. Whether the Red Bird hits the structure, the toss is too short, or the toss is too long.

**Code**

**Meters Code:**

#include <iostream>

#include<math.h>

using namespace std;

int main()

{

const double a = -9.81;

const double PI = 3.14159;

double theta;

double phi;

double BirdHeight;

double Height;

double Distance;

double velocityX;

double velocityY;

double FVY;

double FVX;

double Angle;

double FinalBirdVelocity;

double v;

double t;

cout << "Enter the velocity in m/s: ";

cin >> v;

cout << "Enter the Horizontal Target Distance in Meters: ";

cin >> Distance;

cout << "Enter theta in degrees: ";

cin >> theta;

cout << "Enter the height of structure in meters: ";

cin >> Height;

phi = theta \* (PI / 180);

velocityX = v \* cos(phi);

velocityY = v \* sin(phi);

FVX = velocityX;

t = (Distance / velocityX);

BirdHeight =(velocityY\*t)+((1/2)\*(a)\*(t\*t));

FVY = sqrt(fabs((velocityY\*velocityY) + ((2 \* a)\*(Height - 5))));

FinalBirdVelocity = sqrt((FVX\*FVX) + (FVY\*FVY));

Angle = atan(FVY / FVX)\*(180/PI);

cout << "The final velcoity is: "<<FinalBirdVelocity<<"m/s"<<endl;

cout << "The time to reach strucutre is:" << t << "s" << endl;

cout << "The dirrection is: " << Angle << "degrees" << endl;

cout << "The height of the bird is: " << BirdHeight << "m" << endl;

if (BirdHeight <= Height) {

cout << "The Bird has hit the Tower" << endl;

}

else {

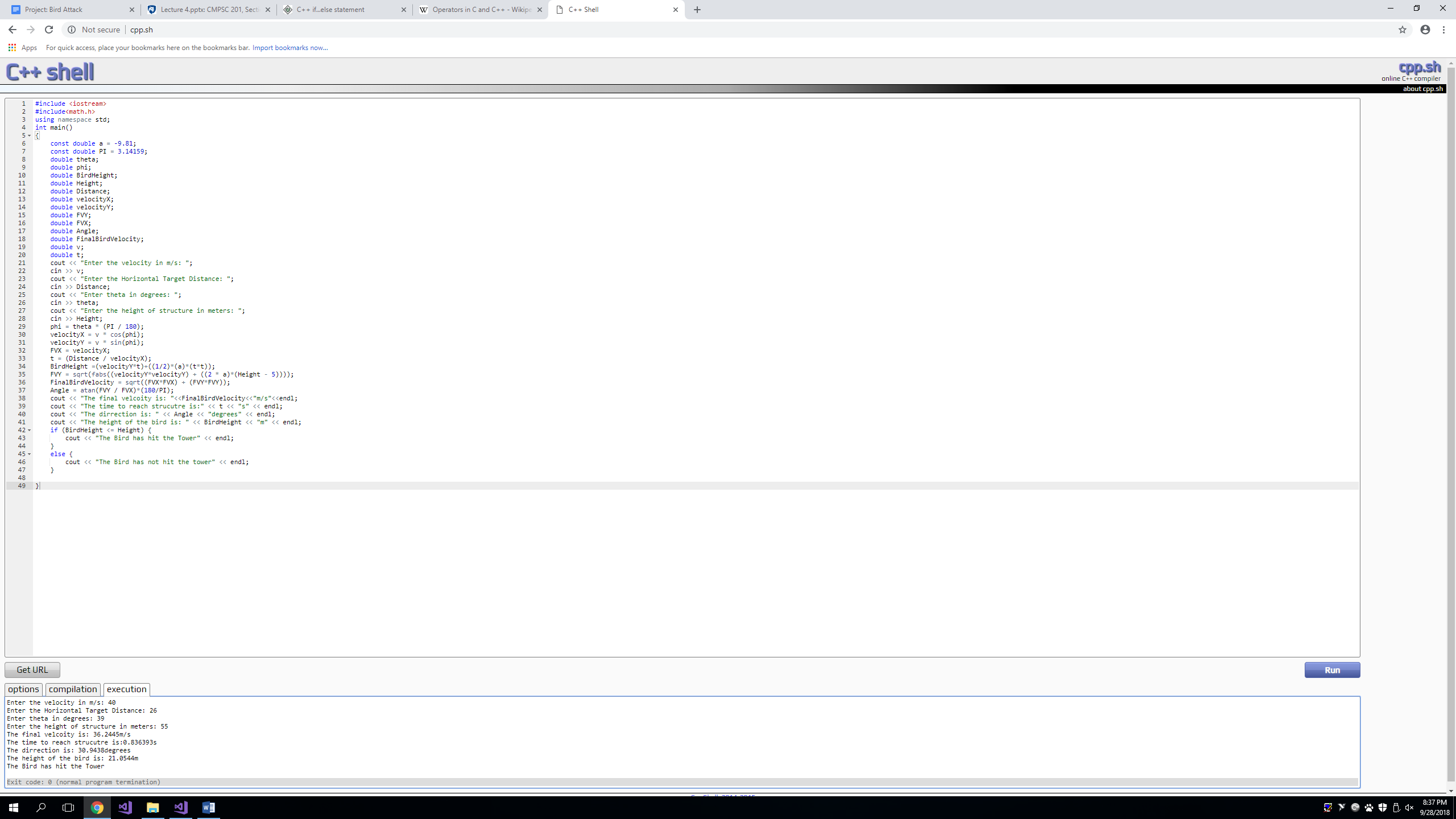
cout << "The Bird has not hit the tower" << endl;

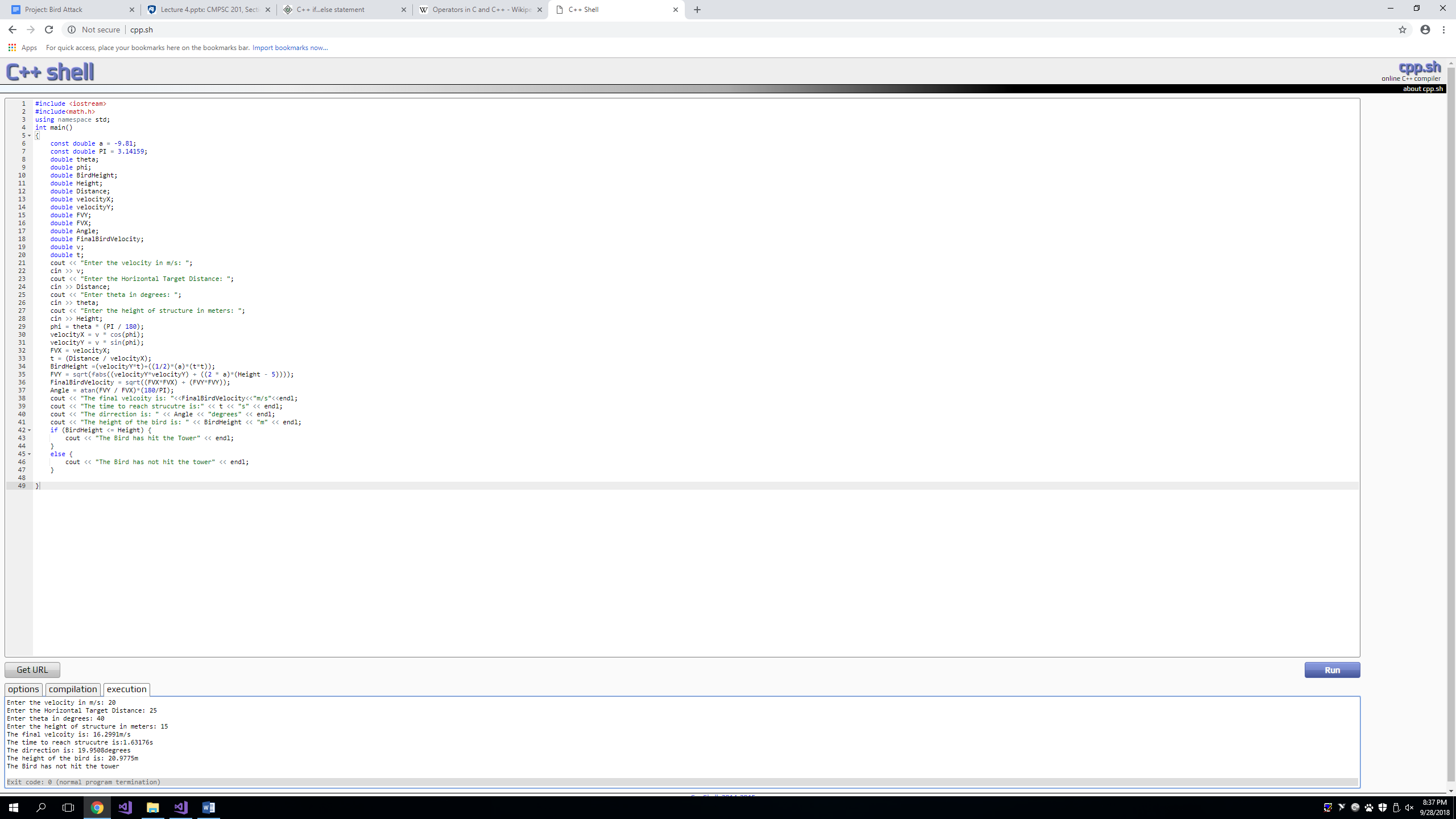
}

return 0;

}

**Sample Runs**





**Code**

#include <iostream>

#include<math.h>

using namespace std;

int main()

{

const double a = -32;

const double PI = 3.14159;

double theta;

double phi;

double BirdHeight;

double Height;

double Distance;

double velocityX;

double velocityY;

double FVY;

double FVX;

double Angle;

double FinalBirdVelocity;

double v;

double t;

cout << "Enter the velocity in ft/s: ";

cin >> v;

cout << "Enter the horizontal target distance in feet: ";

cin >> Distance;

cout << "Enter theta in degrees: ";

cin >> theta;

cout << "Enter the height of structure in feet: ";

cin >> Height;

phi = theta \* (PI / 180);

velocityX = v \* cos(phi);

velocityY = v \* sin(phi);

FVX = velocityX;

t = (Distance / velocityX);

BirdHeight = (velocityY\*t) + ((1 / 2)\*(a)\*(t\*t));

FVY = sqrt(fabs((velocityY\*velocityY) + ((2 \* a)\*(Height - 5))));

FinalBirdVelocity = sqrt((FVX\*FVX) + (FVY\*FVY));

Angle = atan(FVY / FVX)\*(180 / PI);

cout << "The final velcoity is: " << FinalBirdVelocity << "ft/s" << endl;

cout << "The time to reach strucutre is:" << t << "s" << endl;

cout << "The dirrection is: " << Angle << "degrees" << endl;

cout << "The height of the bird is: " << BirdHeight << "ft" << endl;

if (BirdHeight <= Height) {

cout << "The Bird has hit the Tower" << endl;

}

else {

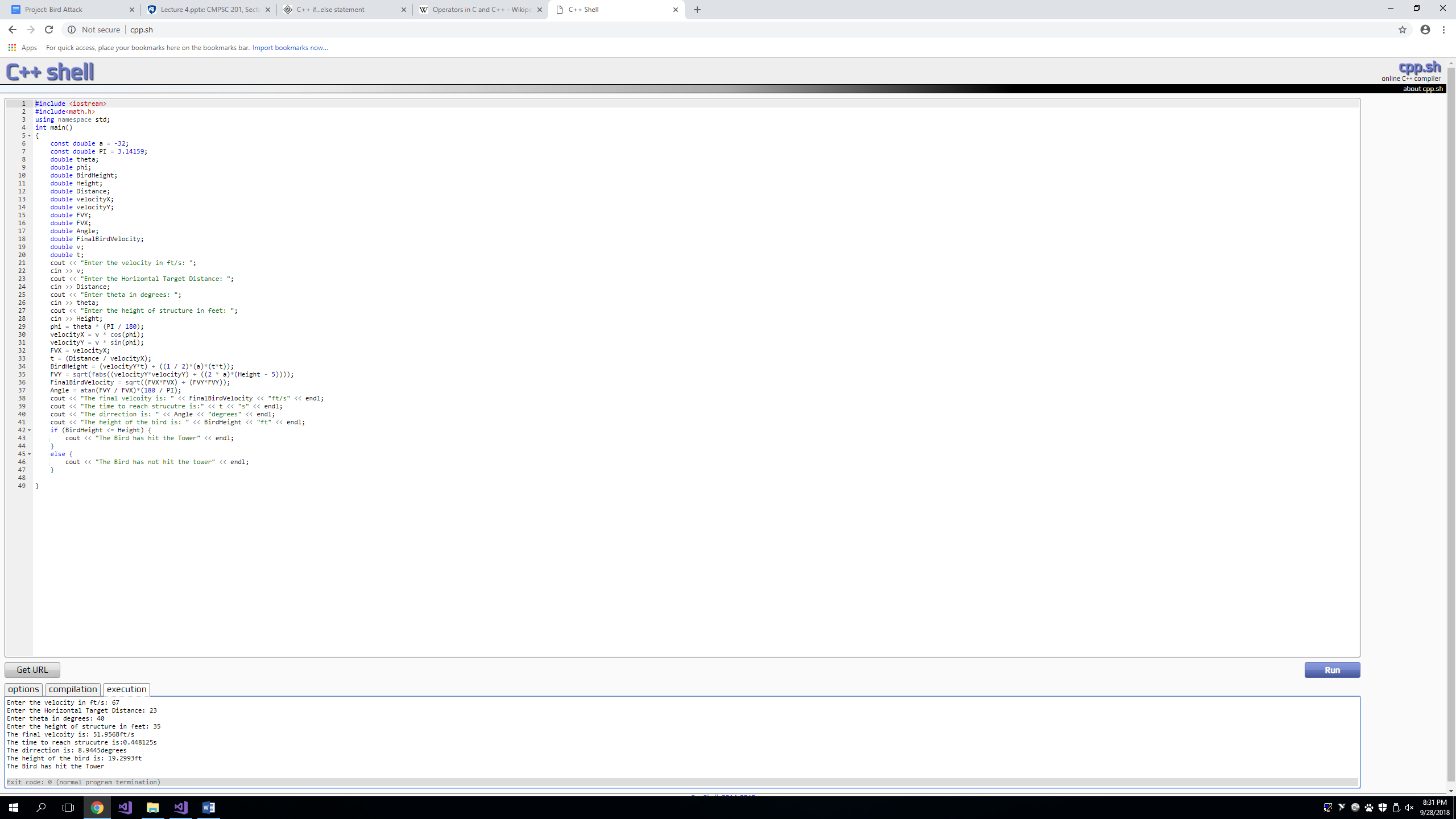
cout << "The Bird has not hit the tower" << endl;

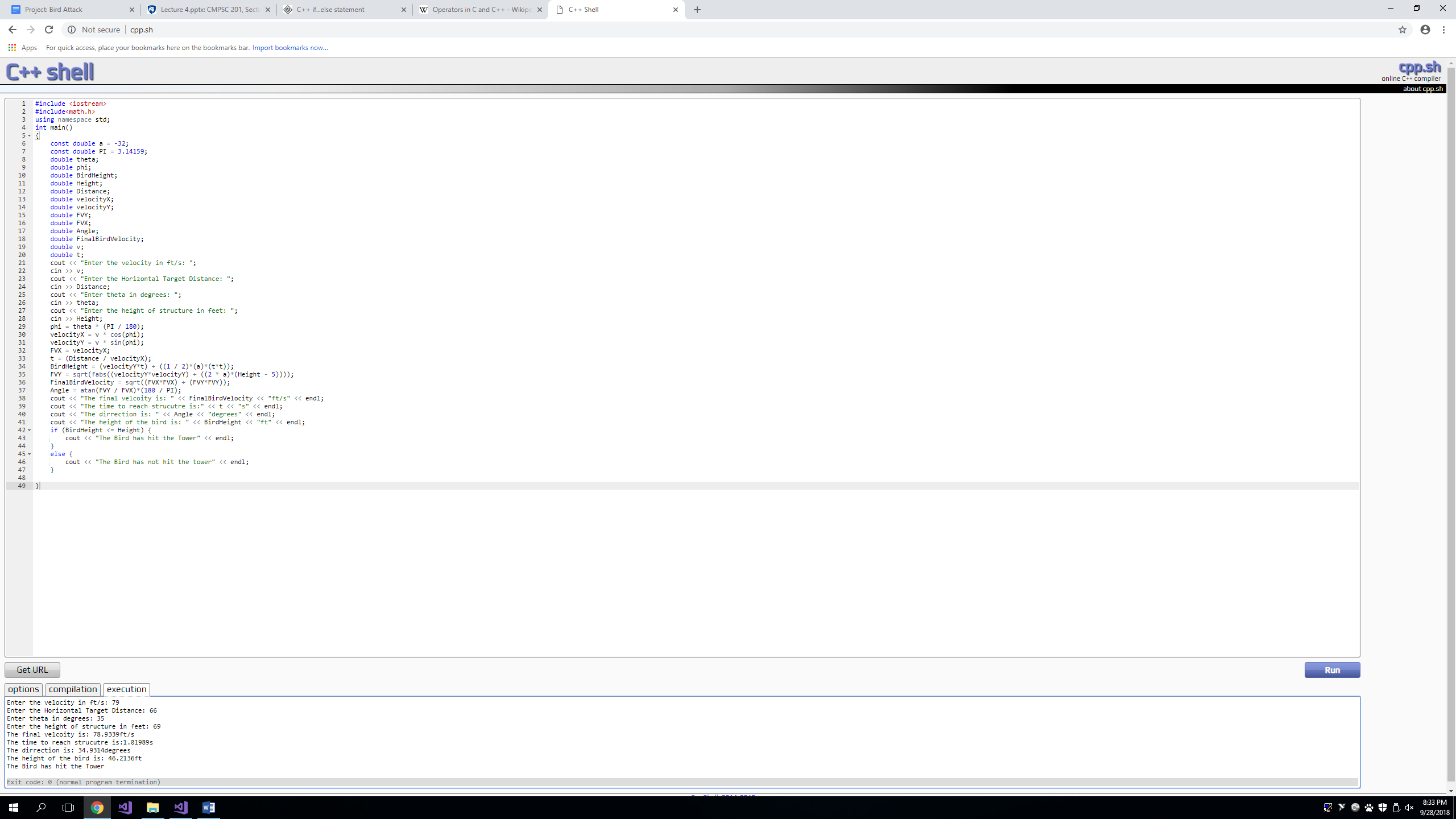
}

return 0;

}

**Sample Runs**





**Discussion**

1. How many hours did this program take?
2. What type of problems did you run into while programming this Project? How did you resolve them?
3. How could you enhance the program?
4. This program has taken at least seven hours in order to entirely write the code, debug it, and adjust the variables.
5. I ran into the problem of having strange outputs, the program was spitting out extremely large and odd numbers and even running into errors with having a final velocity and final angle. It turned out that I had to use the absolute values in FXY in order to have a positive value for the square root to compute the number without errors. Also assigning variables to certain values was troubling, it was a learning curve; however, it was manageable towards the last few hours of finishing the code. Some of the mathematical functions were not very defined on how to use but using the PowerPoints, it made more sense how some of those functions were being applied in the code. Also another problem was determining which variable is an integer or “double agent”; however, it turned out that the solution was making all the variables besides the constants as “double agents”.
6. The program could be enhanced if it was trimmed down as some steps in the program are for safety measures. The structure of the code could have been better but due to the many changes that happened at the beginning of the code. Also the time efficiency, I could have utilized my time better reading the code and seeing what is wrong instead of trying multiple test runs to see what is wrong.