

Wave Generator Readme

By Seriki, Ayodele Hassanat

170408008

B.Sc. Computer Engineering

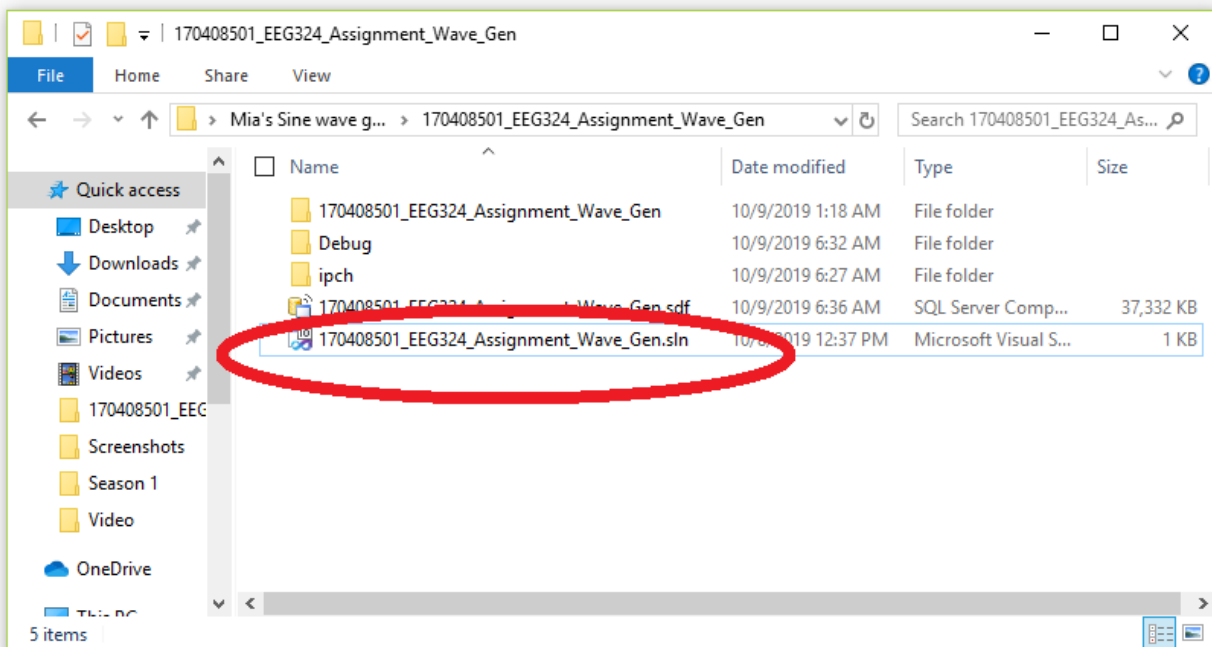
Department of Electrical and Electronics Engineering

University of Lagos

Step 1

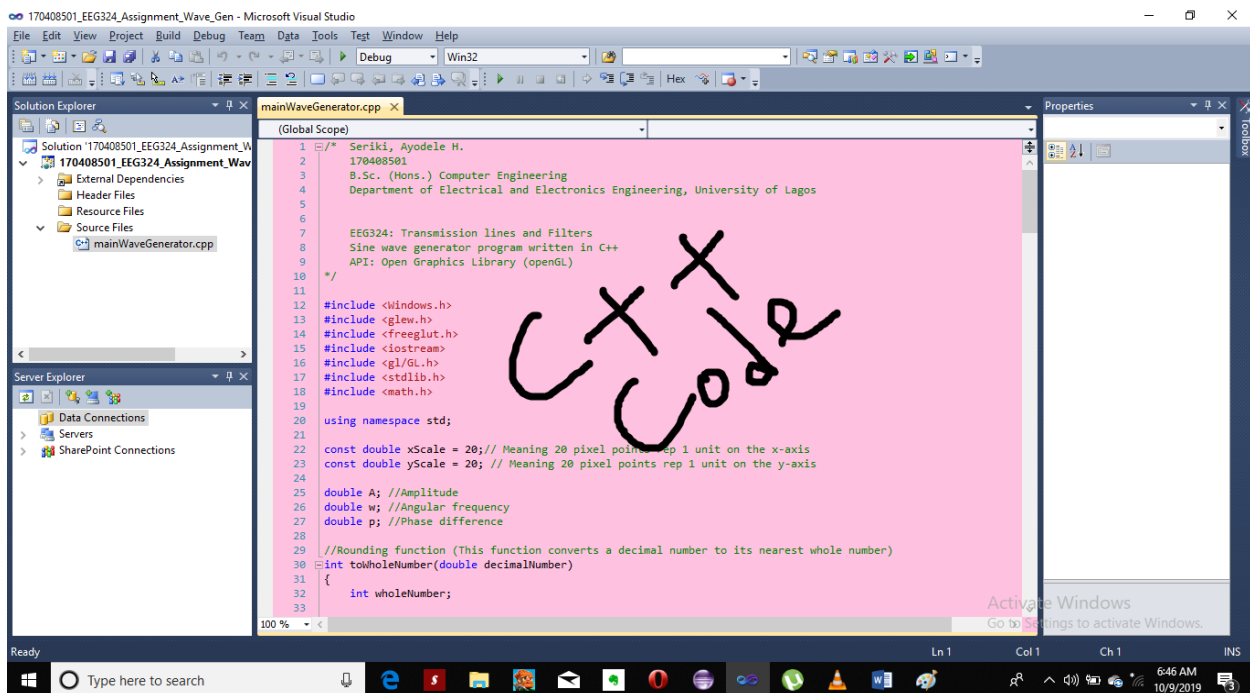
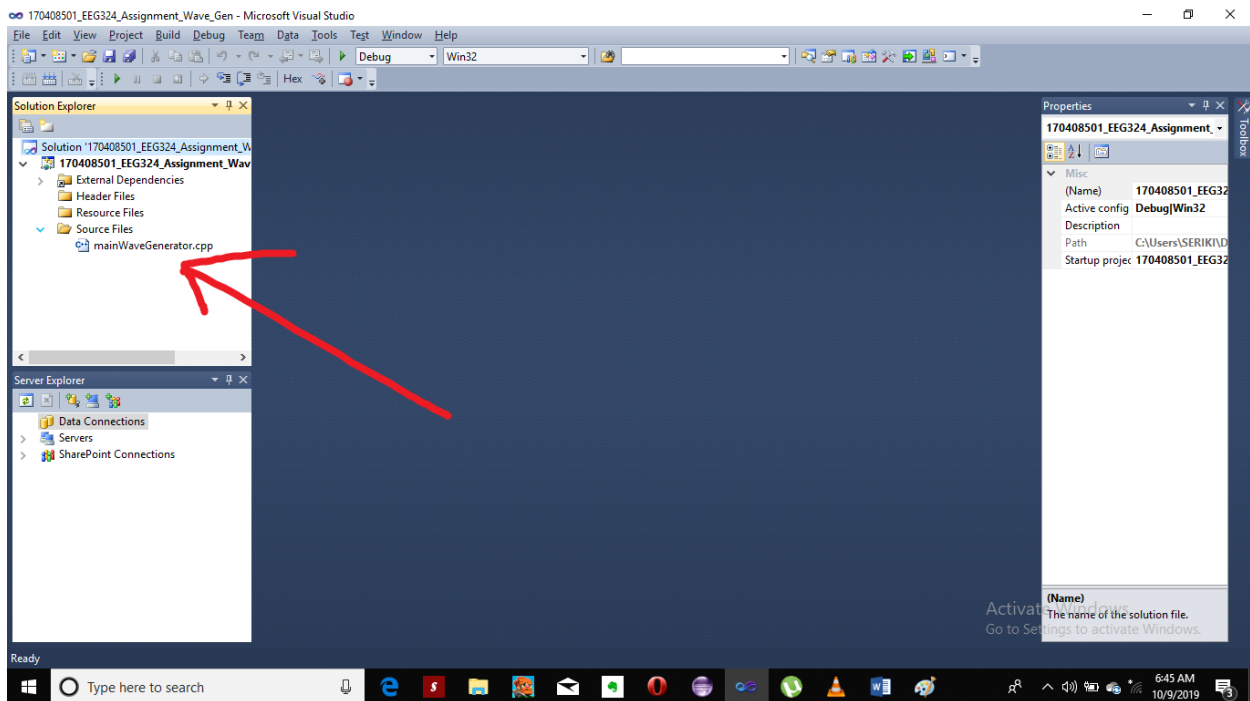
Extract the whole file contents of

170408501_EEG324_Assignment_Wave_Gen.zip and open the Microsoft Visual Studio solution file, **170408501_EEG324_Assignment_Wave_Gen.sln**.



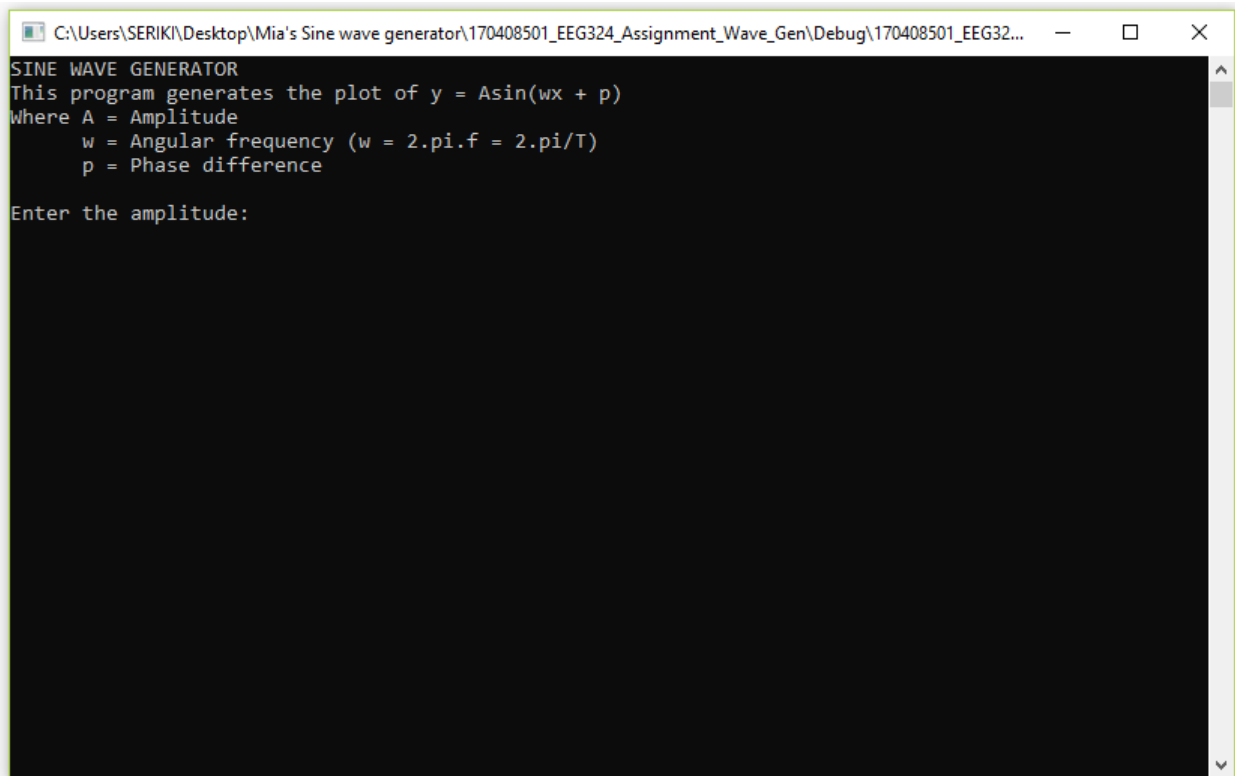
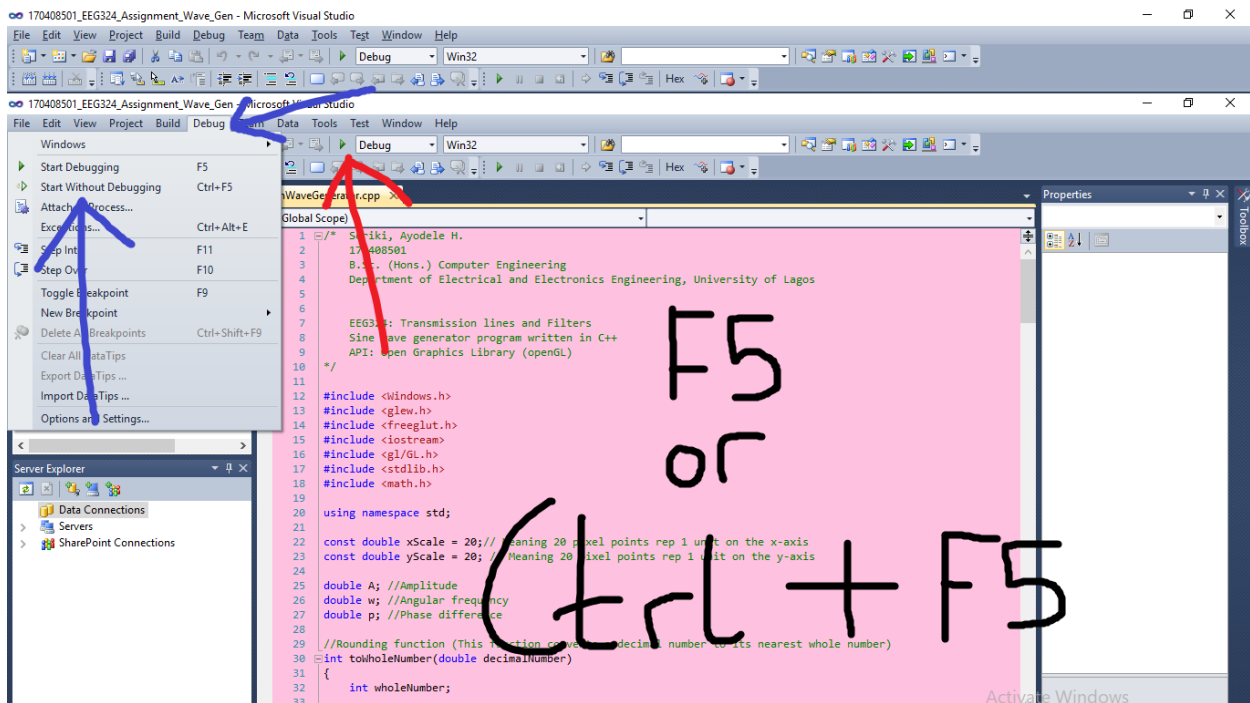
Step 2

Navigate to the **source files** directory in the solution explorer and double-click **mainWaveGenerator.cpp** to see the code.



Step 3

To run the program, click **Debug** on the menu bar then click on either **Start Debugging** or **Start Without Debugging** or click on The green 'play' button on the toolbox to debug or just press **f5** or **ctrl + f5**.



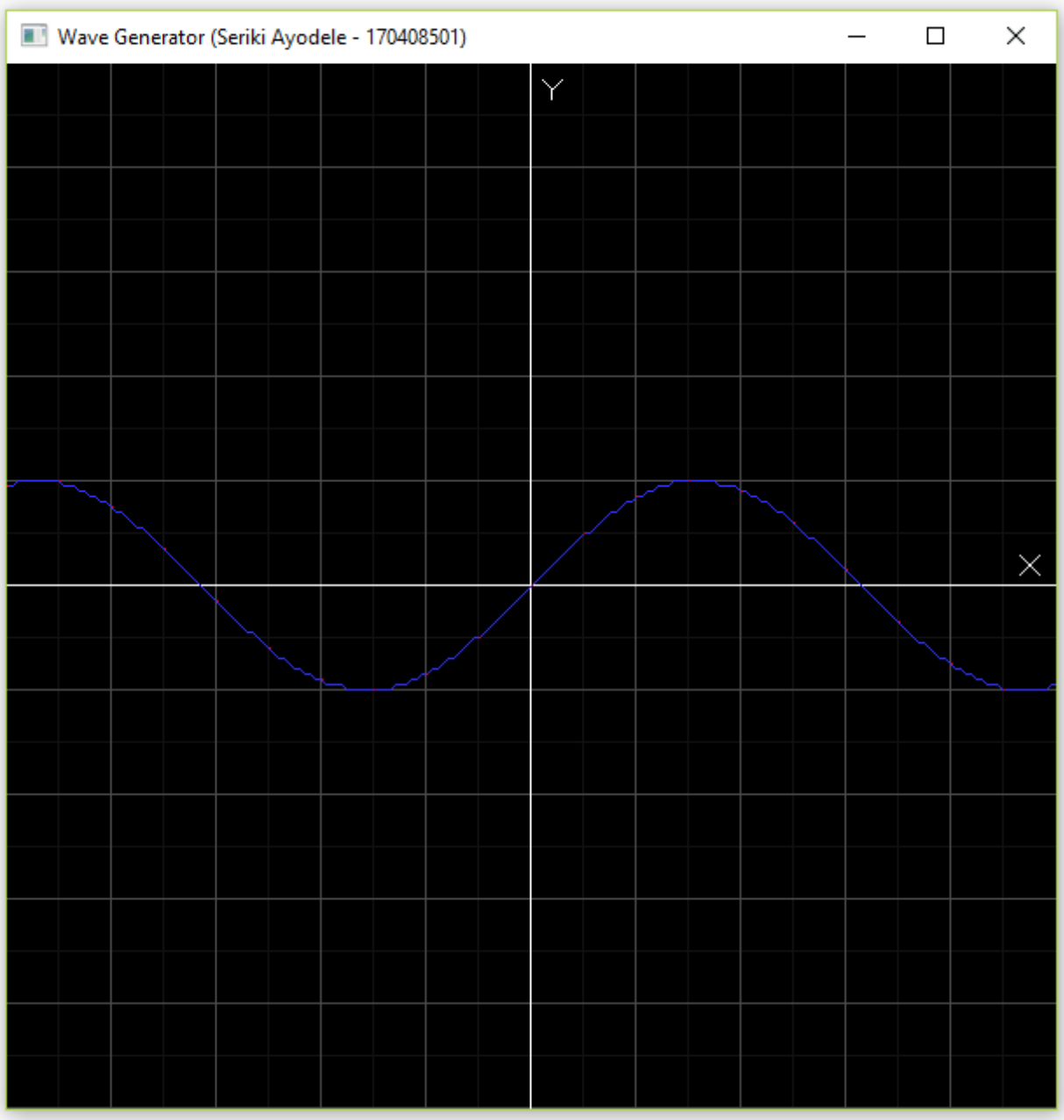
Step 4

Enter the values of the amplitude, angular frequency and phase of the sine wave you wish to generate.

```
C:\Users\SERIKI\Desktop\Mia's Sine wave generator\170408501_EEG324_Assignment_Wave_Gen\Debug\170408501_EEG32...
SINE WAVE GENERATOR
This program generates the plot of  $y = A\sin(wx + p)$ 
Where A = Amplitude
      w = Angular frequency ( $w = 2\pi \cdot f = 2\pi/T$ )
      p = Phase difference

Enter the amplitude: 1
Enter the angular frequency: 1
Enter the phase difference: 0
The wave  $y = 1\sin(1x + 0)$  is being generated
```

$$y = \sin x$$



Generated wave

C++ Code

```
/*      Seriki, Ayodele H.
      170408501
      B.Sc. (Hons.) Computer Engineering
      Department of Electrical and Electronics Engineering, University of Lagos

      EEG324: Transmission lines and Filters
      Sine wave generator program written in C++
      API: Open Graphics Library (OpenGL)
*/

#include <Windows.h>
#include <glew.h>
#include <freeglut.h>
#include <iostream>
#include <gl/GL.h>
#include <stdlib.h>
#include <math.h>

using namespace std;

const double xScale = 20; // Meaning 20 pixel points rep 1 unit on the x-axis
const double yScale = 20; // Meaning 20 pixel points rep 1 unit on the y-axis

double A; //Amplitude
double w; //Angular frequency
double p; //Phase difference

//Rounding function (This function converts a decimal number to its nearest whole number)
int toWholeNumber(double decimalNumber)
{
    int wholeNumber;

    if(decimalNumber >= 0)
    {
        wholeNumber = (int)(decimalNumber + 0.5);
    }
    else
    {
        decimalNumber *= -1;
        wholeNumber = (int)(decimalNumber + 0.5);
        wholeNumber *= -1;
    }

    return wholeNumber;
}

void display(void)
{
    glBegin(GL_LINES);
    {
        //Grey (Darker)
        glColor3f(0.07, 0.07, 0.07);
```

```

//0.5 unit lines (vertical)
for (int k = 0; k < 20; k++)
{
    glVertex2s(100, (-100 + k*10));
    glVertex2s(-100, (-100 + k*10));
}

//0.5 unit lines (horizontal)
for (int k = 0; k < 20; k++)
{
    glVertex2s((-100 + k*10), 100);
    glVertex2s((-100 + k*10), -100);
}

//Grey (Lighter)
glColor3f(0.3, 0.3, 0.3);

//1 unit lines (vertical)
for (int k = 0; k < 10; k++)
{
    glVertex2s(100, (-100 + k*20));
    glVertex2s(-100, (-100 + k*20));
}

//1 unit lines (horizontal)
for (int k = 0; k < 10; k++)
{
    glVertex2s((-100 + k*20), 100);
    glVertex2s((-100 + k*20), -100);
}

//White
glColor3f(1.0, 1.0, 1.0);

//X-axis line
glVertex2s(100, 0);
glVertex2s(-100, 0);

//Y-axis line
glVertex2s(0, 100);
glVertex2s(0, -100);
}
glEnd();

glBegin(GL_LINES);
{
    //White
    glColor3f(1.0, 1.0, 1.0);

    //X label
    glVertex2s(93, 2);
    glVertex2s(97, 6);
    glVertex2s(97, 2);
    glVertex2s(93, 6);

    //Y label

```

```

        glVertex2s(4, 95);
        glVertex2s(6, 97);
        glVertex2s(4, 95);
        glVertex2s(2, 97);
        glVertex2s(4, 95);
        glVertex2s(4, 93);
    }
    glEnd();

    glBegin(GL_LINE_STRIP);
    {
        glColor3f(0.2, 0.2, 1.0);

        int x_plot, y_plot;

        //Sine wave plotter
        for (int l = -100; l <= 100; l++)
        {
            x_plot = l;
            y_plot = toWholeNumber(A*yScale*(sin(((double)(l*w)/xScale) + p)));
            glVertex2s(x_plot, y_plot);
        }
    }
    glEnd();

    glBegin(GL_POINTS);
    {
        glColor3f(1.0, 0.0, 0.0);

        int x_plot, y_plot;

        //Sine wave plotter
        for (int l = -100; l <= 100; l+=10)
        {
            x_plot = l;
            y_plot = toWholeNumber(A*yScale*(sin(((double)(l*w)/xScale) + p)));
            glVertex2s(x_plot, y_plot);
        }
    }
    glEnd();

    glFlush();
}

void init(void)
{
    glClearColor(0.0, 0.0, 0.0, 0.0);
    glOrtho(-100.0, 100.0, -100.0, 100.0, -1.0, 1.0);
}

int main(int argc, char** argv)
{
    cout << "SINE WAVE GENERATOR" << endl;
    cout << "This program generates the plot of  $y = A\sin(wx + p)$ " << endl;
    cout << "Where A = Amplitude" << endl;
    cout << "      w = Angular frequency ( $w = 2\pi \cdot f = 2\pi/T$ )" << endl;
}

```



```

    cout << "          p = Phase difference" << endl << endl;

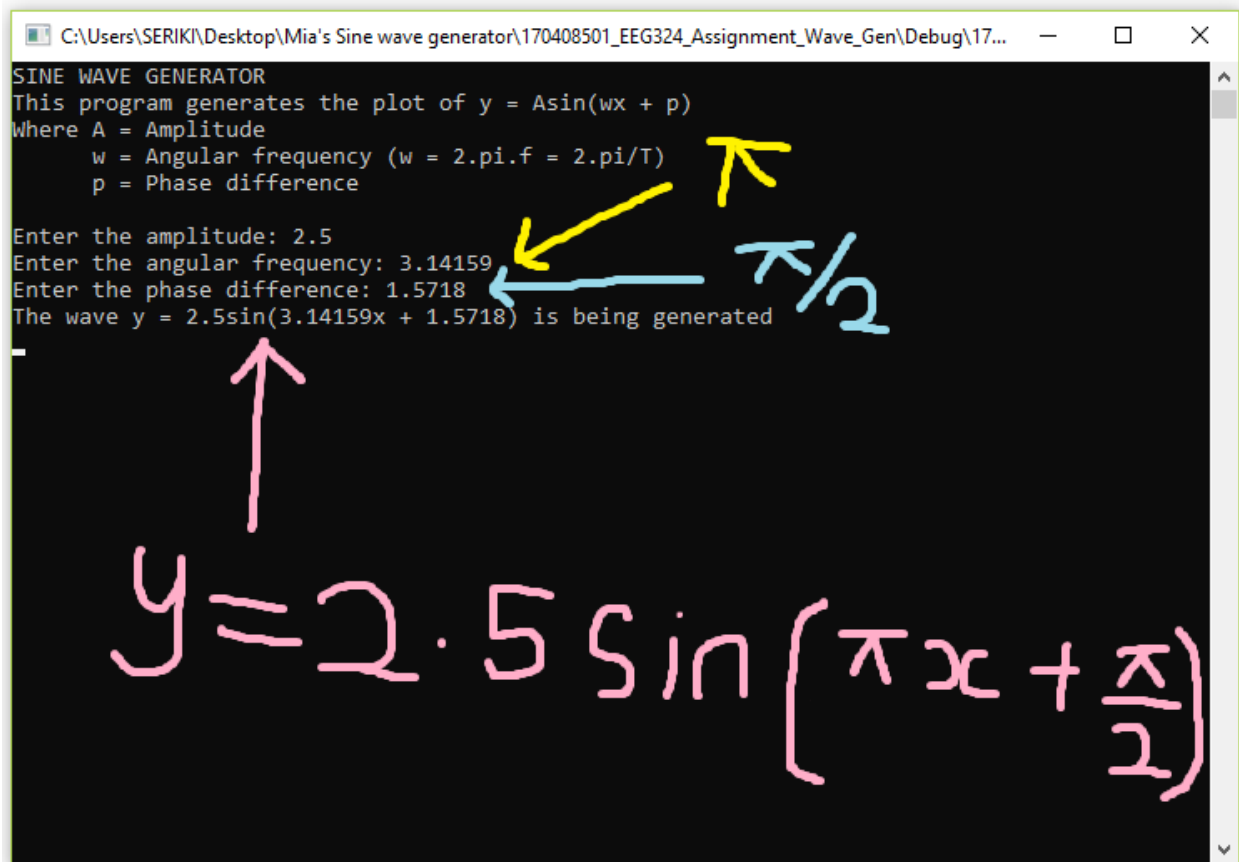
    cout << "Enter the amplitude: ";
    cin >> A;
    cout << "Enter the angular frequency: ";
    cin >> w;
    cout << "Enter the phase difference: ";
    cin >> p;

    //Description of the final equation to be plotted
    cout << "The wave y = " << A << "sin(" << w << "x + " << p << ") is being
generated" << endl;

    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(600, 600);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("Wave Generator (Seriki Ayodele - 170408501)");
    init();
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}

```

Output (Another Example)



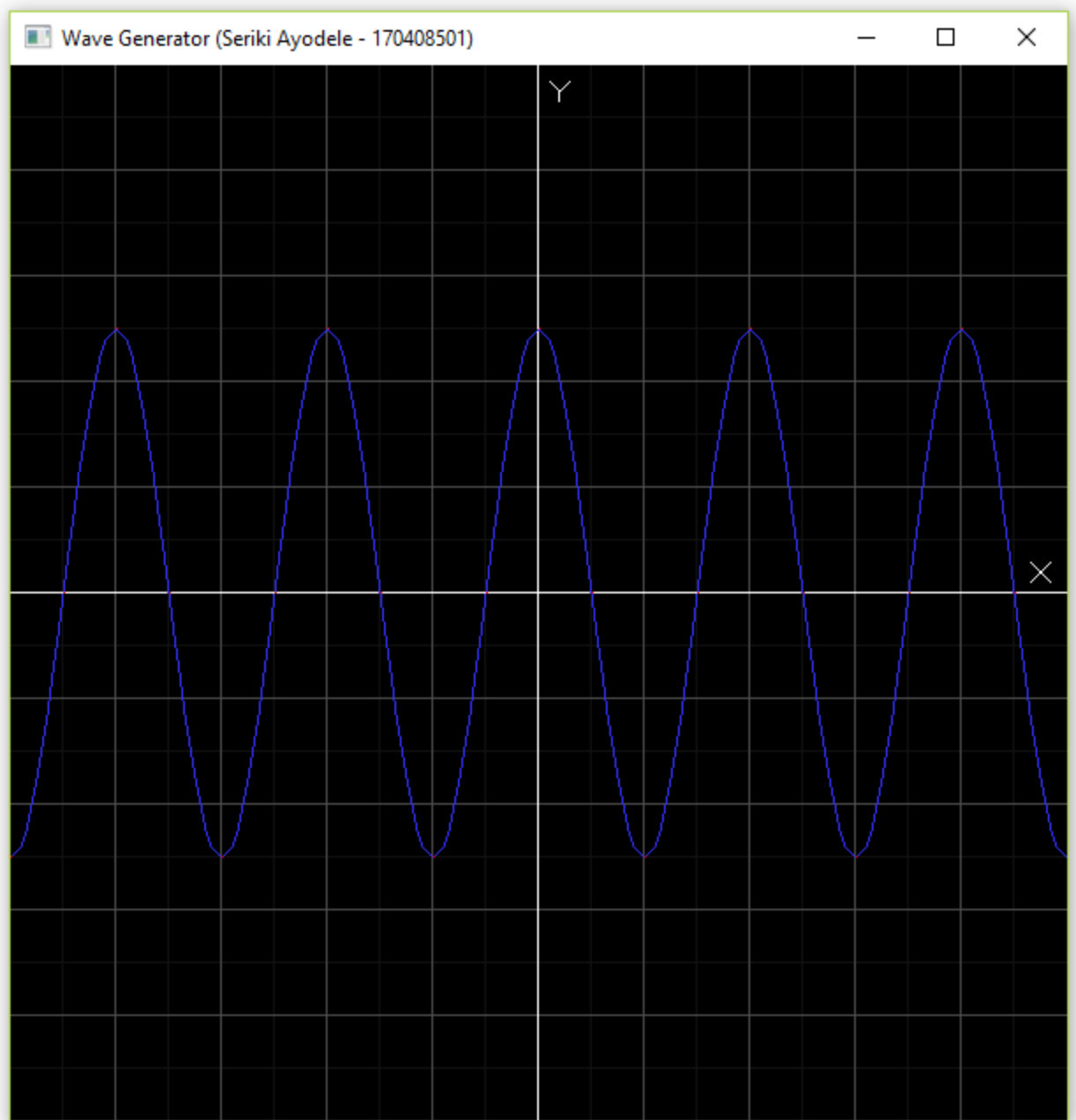
```
SINE WAVE GENERATOR
This program generates the plot of  $y = A\sin(wx + p)$ 
Where A = Amplitude
      w = Angular frequency ( $w = 2\pi f = 2\pi/T$ )
      p = Phase difference

Enter the amplitude: 2.5
Enter the angular frequency: 3.14159
Enter the phase difference: 1.5718
The wave  $y = 2.5\sin(3.14159x + 1.5718)$  is being generated
```

Handwritten annotations:

- Yellow arrows point from the text "Angular frequency" and "Phase difference" to the values 3.14159 and 1.5718 respectively.
- A blue arrow points from the value 3.14159 to the handwritten π in the equation.
- A pink arrow points from the value 1.5718 to the handwritten $\frac{\pi}{2}$ in the equation.
- The final equation $y = 2.5\sin(\pi x + \frac{\pi}{2})$ is written in pink.

$$y = 2.5 \sin\left(\pi x + \frac{\pi}{2}\right)$$



Generated wave