Computer Graphics
Assignment 1

Midpoint Line Drawing using integer arithmetic:

In this the algorithm is divided into 4 cases:

Case 1: When slope = 0 then draw a straight line from x0 to x1 keeping y constant.

Case 2: When slope = infinite draw a straight line form y0 to y1 keeping x constant.

Case 3: when slope > 0 use the algorithm defined as,

```
dx=x1-x0;
dy=y1-y0;
d=2*dy - dx;
de=2*dy;
dne = 2*(dy-dx);
x=x0;
y=y0;
draw(x, y);
while (x<=x1)
{
   if (d<=0)
   {
      d = d + de;
      x++;
   }
   else
   {
      d = d + dne;
      x++;
      y++;
   }
draw(x, y);
```

Case 4: when slope < 0 use the algorithm defined as,

```
dx=x1-x0;
dy=y1-y0;
d=-2*dy - dx;
de=-2*dy;
dne = -2*(dy+dx);
x=x0;
y=y0;
draw(x, y);
while (x<=x1)
{
   if (d<=0)
   {
      d = d + de;
      x++;
   }
```

```
          else
          {
             d = d + dne;
             x++;
             y--;
          }
       draw(x, y);
```

Triangle Drawing:

In this the code is divided into some sections as:

1. Code for Drawing the triangle edges(initially commented if edges are to be shown remove the comments from the function draw_edges()).
2. Code for drawing the Bounding box.
3. Code for drawing the triangle.
4. Super Sampling to remove the jaggering, in this each pixel is and its neighbouring pixel is taken into account then the percentage is calculated as (the number of pixels inside the triangle)/2, and with the percentage the color is filled in the pixel.

Circle Drawing:

- Radius = 100
- Center = 200,200
- Starting from 135 to 215 degrees which can be interpreted as 135 to 180 degrees that is 45 degrees and 180 to 215 that is 35 degrees.
- The code is divided into two sections one for 45 degrees that is 135 to 180 and other 35 degrees from 180 to 215. In both the sections the circle is drawn using mid point integer arithmetic and in counter clockwise direction. 8-way symmetry is not used. The window size is defined as 400,400.
- Since integer arithmetic is used for 45 degrees start point is taken as (-0.70*r,0.70*r) and end point is taken as (-r,0) where 0.70 is sin(45). Similarly for 35 degrees start point is taken as (-r,0) and end point as (-0.81*r,-0.57*r) since sin(35)=0.5735.

To Run the code:

Step 1: Install codeblocks in windows and add the GLUT library as defined in the document provided.
Step 2: Create a new GLUT Project as shown in the document.
Step 3: Copy the contents of any one of Line_midpoint.cpp, Circle_midpoint.cpp or Triangle.cpp which you wish to run then paste that content in the main.cpp file of the project you create in step 2 then run the file.