1. Creating a Window

***Note : By default, the window has a size of (300, 300) pixels, and its position is up***

***to the window manager to choose.***

#include<GL/glut.h>

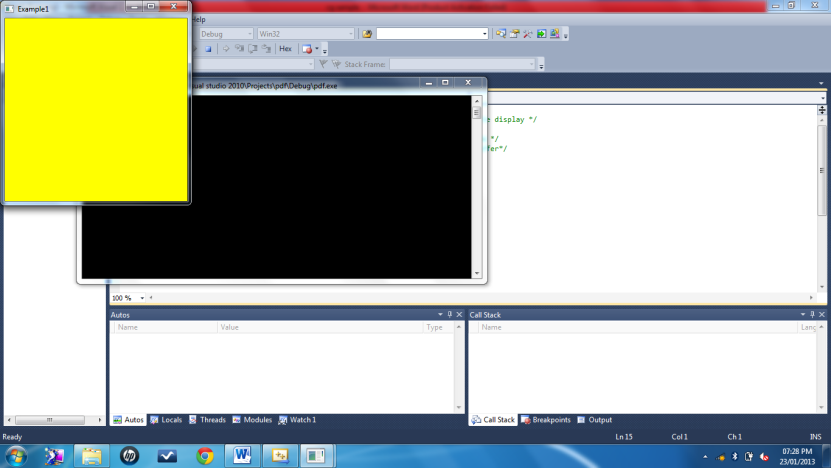
void display () /\* callback function which is called when OpenGL needs to update the display \*/

{

glClearColor (1.0,1.0,0.0,0.0);/\*default color –black…….. Now set to YELLOW \*/

glClear (GL\_COLOR\_BUFFER\_BIT);/\*Clear the window-set the color of pixels in buffer\*/

glFlush(); /\* Force update of screen \*/

}

void main (int argc, char \*\*argv)

{

glutInit (&argc, argv); /\* Initialise OpenGL \*/

glutCreateWindow ("Example1"); /\* Create the window \*/

glutDisplayFunc (display); /\* Register the "display" function \*/

glutMainLoop (); /\* Enter the OpenGL main loop \*/

}

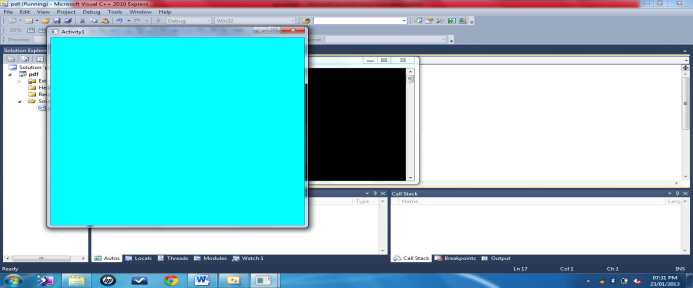
Activity 1

**Change the window size to 500, 500**

**Change the window position to 100, 100**

**Change the window color to CYAN**

#include<GL/glut.h>

void display (void)

{

**glClearColor (0.0,1.0,1.0,0.0);**

glClear (GL\_COLOR\_BUFFER\_BIT);

glFlush();

}

void main (int argc, char \*\*argv)

{

glutInit (&argc, argv); /\* Initialise OpenGL \*/

**glutInitWindowSize(500,500);**

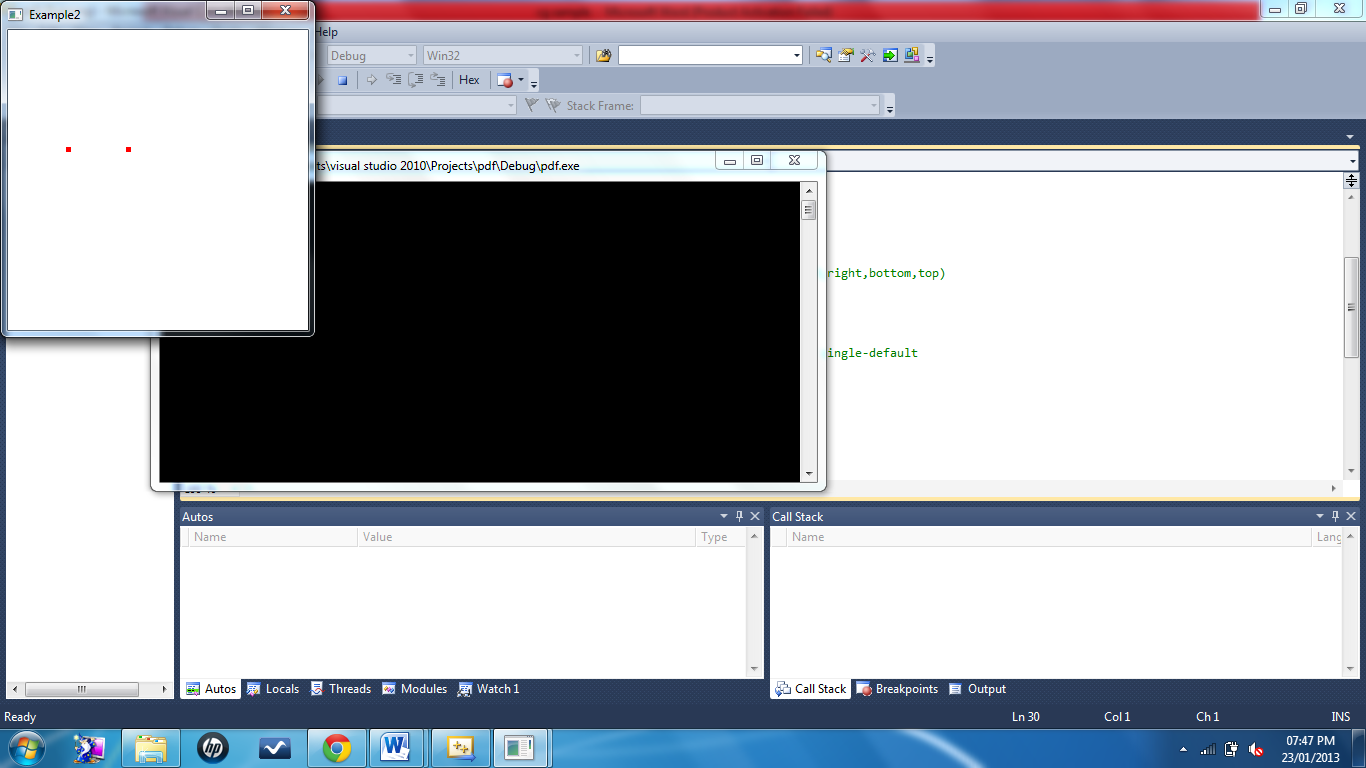
**glutInitWindowPosition(100,100);**

glutCreateWindow ("Activity1"); /\* Create the window \*/

glutDisplayFunc (display); /\* Register the "display" function \*/

glutMainLoop (); /\* Enter the OpenGL main loop \*/

}

2. Drawing pixels/points

#include<GL/glut.h>

void display()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_POINTS);

glVertex2i(100,300);

glVertex2i(201,300);

glEnd();

glFlush();

}

void myinit()

{

glClearColor(1.0,1.0,1.0,1.0);// set the window color to white

glColor3f(1.0,0.0,0.0);// set the point color to red (RGB)

glPointSize(5.0);// set the pixel size

gluOrtho2D(0.0,500.0,0.0,500.0);// coordinates to be used with the viewport(left,right,bottom,top)

}

void main(int argc, char\*\* argv)

{

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);// sets the initial display mode, GLUT single-default

glutInitWindowSize(300,300);

glutInitWindowPosition(0,0);

glutCreateWindow("Example2");

glutDisplayFunc(display);

myinit();

glutMainLoop();

}

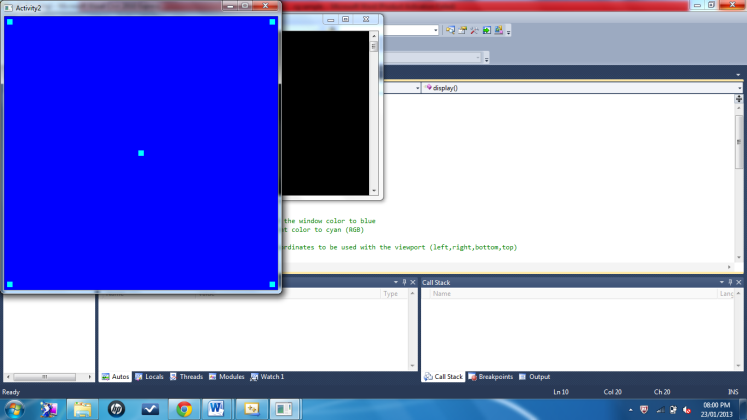
Activity 2

**Change the window color to BLUE**

**Change the point color to CYAN**

**Change the point width to 10**

**Draw FIVE points: at 4 corners of the window and one more at the Centre of the window.**

#include<GL/glut.h>

void display()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_POINTS);

**glVertex2i(10,10);**

**glVertex2i(250,250);**

**glVertex2i(10,490);**

**glVertex2i(490,490);**

**glVertex2i(490,10);**

glEnd();

glFlush();

}

void myinit()

{

**glClearColor(0.0,0.0,1.0,0.0);// set the window color to blue**

**glColor3f(0.0,1.0,1.0);// set the point color to cyan (RGB)**

**glPointSize(10.0);**

gluOrtho2D(0.0,500.0,0.0,500.0);// coordinates to be used with the viewport (left,right,bottom,top)

}

void main(int argc, char\*\* argv)

{

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(500,500);

glutInitWindowPosition(0,0);

glutCreateWindow("Activity2");

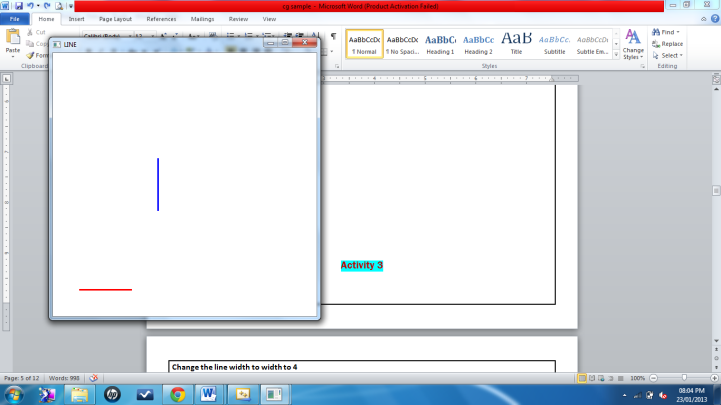
glutDisplayFunc(display);

myinit();

glutMainLoop();

}

3. Drawing lines

#include<GL/glut.h>

void display()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(1.0,0.0,0.0); //draw the line with red color

glLineWidth(3.0); // Thickness of line

glBegin(GL\_LINES);

glVertex2d (50,50); // to draw horizontal line in red color

glVertex2d (150,50);

glColor3f(0.0,0.0,1.0); //draw the line with blue color

glVertex2d (200,200); // to draw vertical line in blue color

glVertex2d (200,300);

glEnd();

glFlush();

}

void myinit()

{

glClearColor(1.0,1.0,1.0,1.0);

glColor3f(1.0,0.0,0.0);

glPointSize(1.0);

gluOrtho2D(0.0,500.0,0.0,500.0);

}

void main(int argc, char\*\* argv)

{

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(500,500);

glutInitWindowPosition(100,100);

glutCreateWindow("LINE");

glutDisplayFunc(display);

myinit();

glutMainLoop();

}

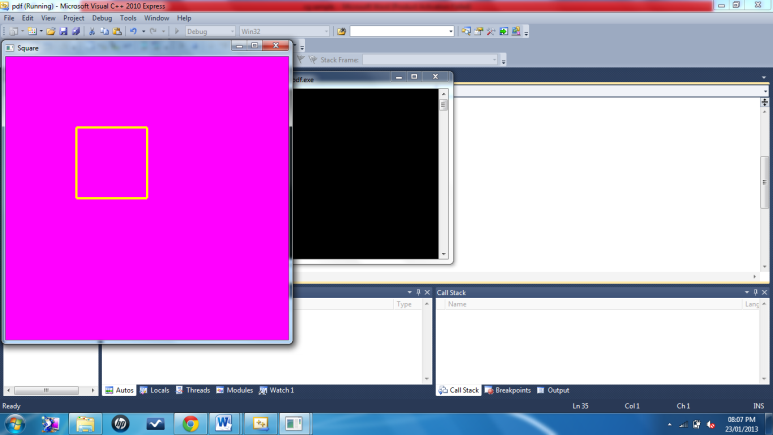
Activity 3

**Change the window color to MAGENTA**

**Change the line color to YELLOW**

**Change the line width to width to 4**

**Draw a square using 4 lines**



#include<GL/glut.h>

void display()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

**glColor3f(1.0,1.0,0.0);**

**glLineWidth(4.0);**

glBegin(GL\_LINES);

**glVertex2d (50, 100);**

**glVertex2d (100, 100);**

**glVertex2d (100, 100);**

**glVertex2d (100, 150);**

**glVertex2d (100, 150);**

**glVertex2d (50, 150);**

**glVertex2d (50, 150);**

**glVertex2d (50, 100);**

glEnd();

glFlush();

}

void myinit()

{

glClearColor(1.0,0.0,1.0,1.0);

gluOrtho2D(0.0,200.0,0.0,200.0);

}

void main(int argc, char\*\* argv)

{

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(500,500);

glutInitWindowPosition(10,100);

glutCreateWindow("Square");

glutDisplayFunc(display);

myinit();

glutMainLoop();

}

4. Drawing a square using LINE\_LOOP

#include<GL/glut.h>

void display()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(0.0, 0.0, 1.0);

glLineWidth(3.0);

glBegin(GL\_LINE\_LOOP); // If you put GL\_LINE\_LOOP, it is only boundary.

glVertex2f(50, 50);

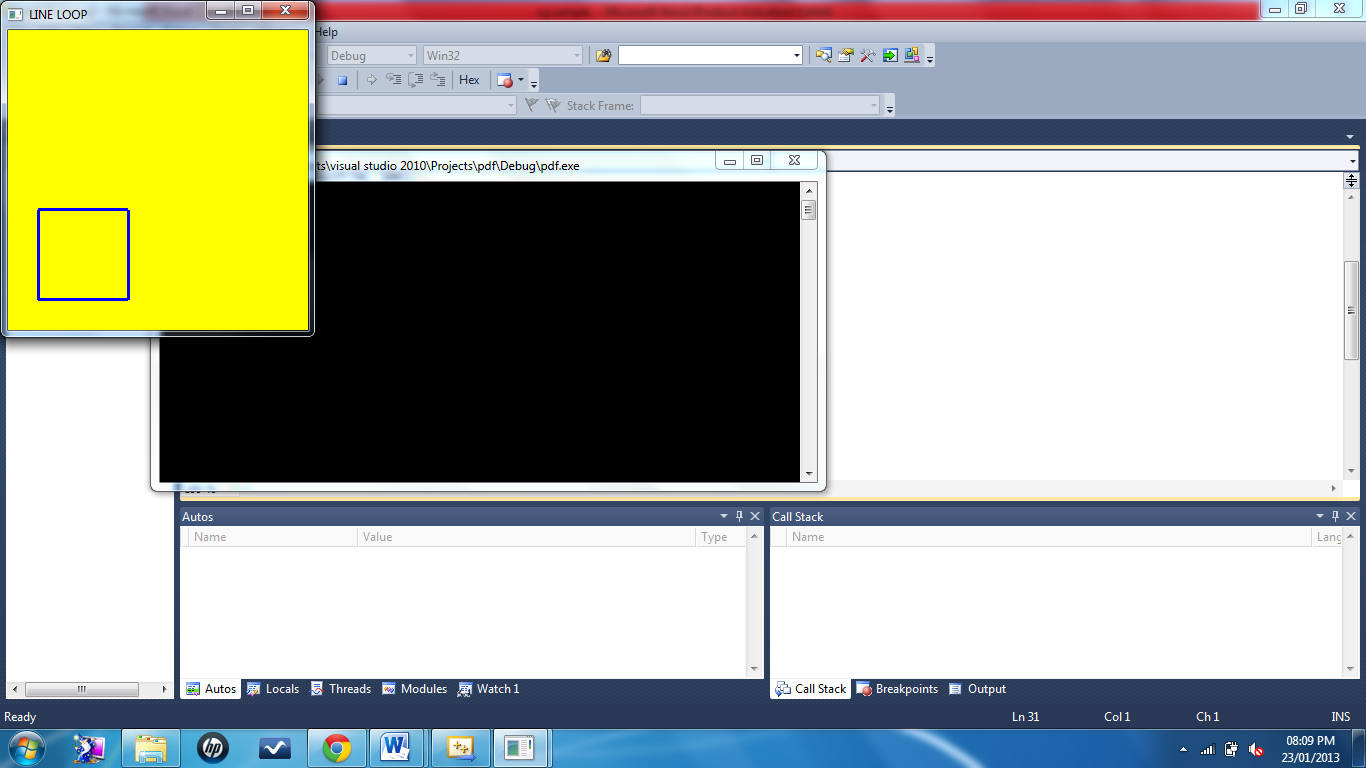
glVertex2f(200, 50);

glVertex2f(200, 200);

glVertex2f(50, 200);

glEnd();

glFlush();

}

void myinit()

{

glClearColor(1.0,1.0,0.0,1.0);

gluOrtho2D(0.0,499.0,0.0,499.0);

}

void main(int argc, char\*\* argv)

{

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(300,300);

glutInitWindowPosition(0,0);

glutCreateWindow("LINE LOOP");

glutDisplayFunc(display);

myinit();

glutMainLoop();

}