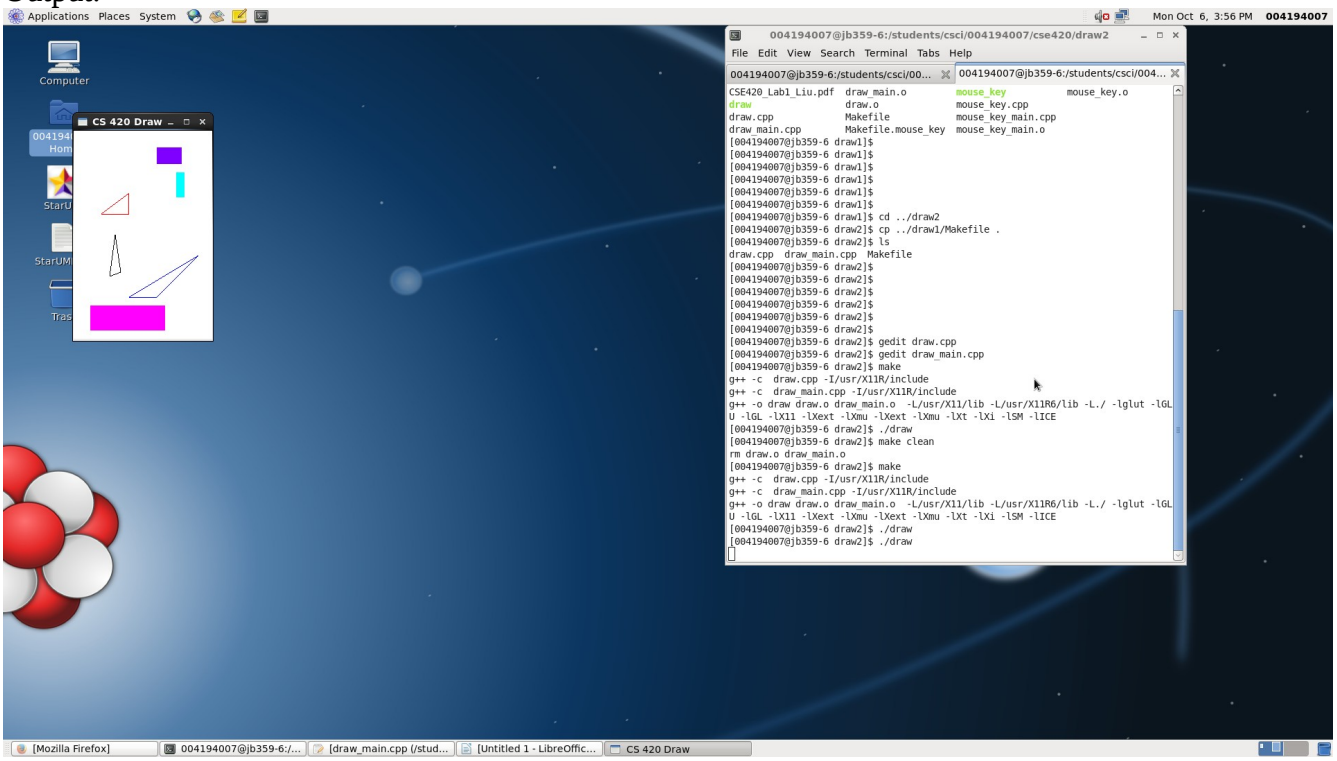
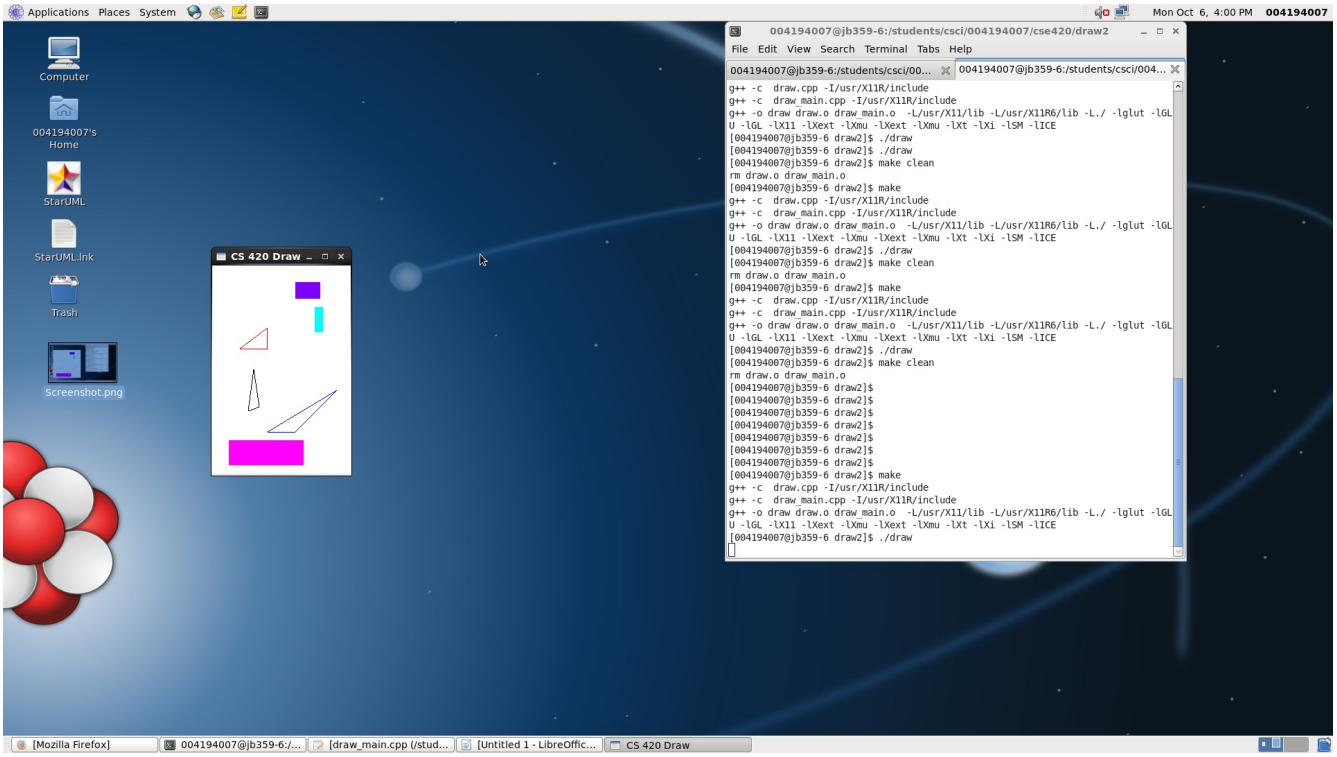
Yazhuo Liu

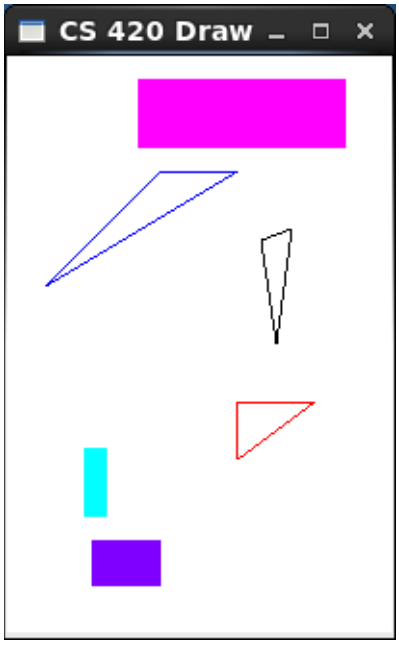
Lab 2
Exercise 1
Output:



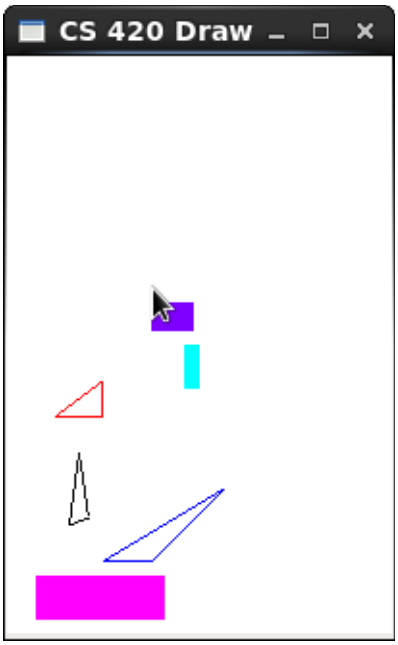
Window size 200x300



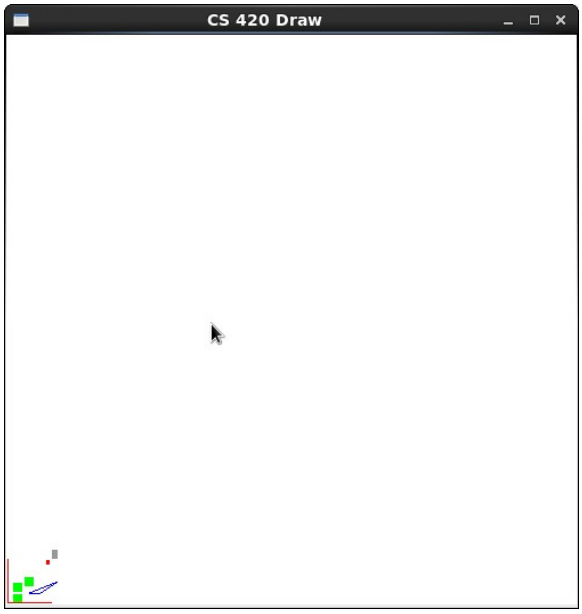
window position 300, 350



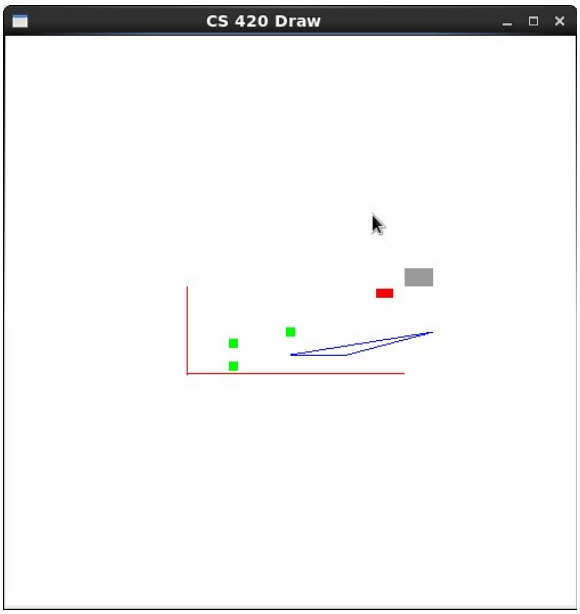
world view (500.0, 0.0, 500.0, 0.0)



world view (0.0, 800.0, 0.0, 800.0)



glViewport(0, 0, 50, 50)



glViewport(150, 200, 250, 100)

Partial codes (because nothing else is changed in the program):

```
//Problem 1 & 2
...
glutInit(&argc, argv); //initialize toolkit
  glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB );   //set display mode: single bufferring, RGB
```

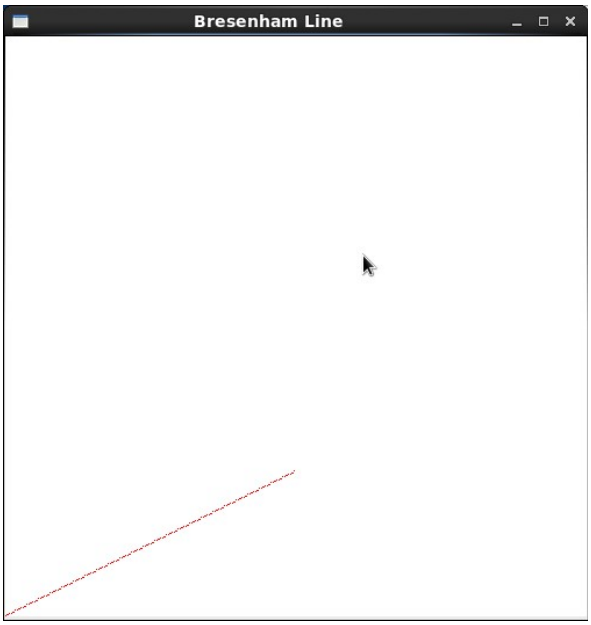
```
model
  glutInitWindowSize(200, 300);               //set window size on screen
  glutInitWindowPosition( 300, 350 );         //set window position on screen
...

//Problem 3:
...
void init( void )
{
  glClearColor( 1.0, 1.0, 1.0, 0.0 );    //get white background color
  glColor3f( 0.0f, 1.0f, 0.0f ); //set drawing color
  glPointSize( 8.0 );                     //a dot is 4x4
  glMatrixMode( GL_PROJECTION );
  glLoadIdentity();                       //replace current matrix with identity matrix
  gluOrtho2D( 0.0, 800.0, 0.0, 800.0 );
}
...

//Problem 4:
...
void display( void )
{
  glViewport( 150, 200, 250, 100 );
  glClear( GL_COLOR_BUFFER_BIT );    //clear screen
  glColor3f ( 0.0, 1.0, 0.0 );
  glBegin( GL_POINTS );                   //draw points
    glVertex2i( 100, 50 );                //draw a point
    glVertex2i( 100, 150 );               //draw a point
    glVertex2i( 200, 200 );               //draw a point
  glEnd();
...
```

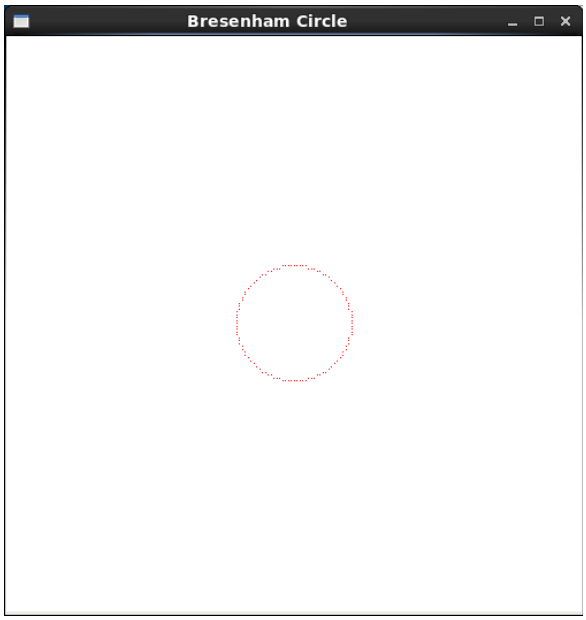
Report:

In problem 1 and 2, I simply just modified the parameters to each function, and it changed their window sizes and window positions. In problem 3, I first changed the function parameters to (500.0, 0.0, 500.0, 0.0), and it changed everything upside down. Then I changed it to (0.0, 800.0, 0.0, 800.0), now the window has a lot of blank space. In problem 4, I added the glViewport() statement, this function gives the window some blank boarding around. I have finished all parts in exercise 1 successfully.

Exercise 2:

Output:





end points (0, 0) and (200, 100)

radius = 20

Partial codes:
//line.cpp

```
...
void line()
{
  int x0 = 0, y0= 0, xn = 200, yn = 100, x, y;
  int     dx, dy,          //deltas
          pk,              //decision parameter
          k;               //looping variable
...
```

//circle.cpp

```
...
void Circle(){

  int xCenter=100,yCenter=100,r=20;
  int x=0,y=r;
  int d = 3/2 - r;                              // = 1 – r
...
```

Report:

I looked the code and instructions, then I found out the solutions to the problems. Also I was not really sure that I had to use the Makefile to compile, but now I do.