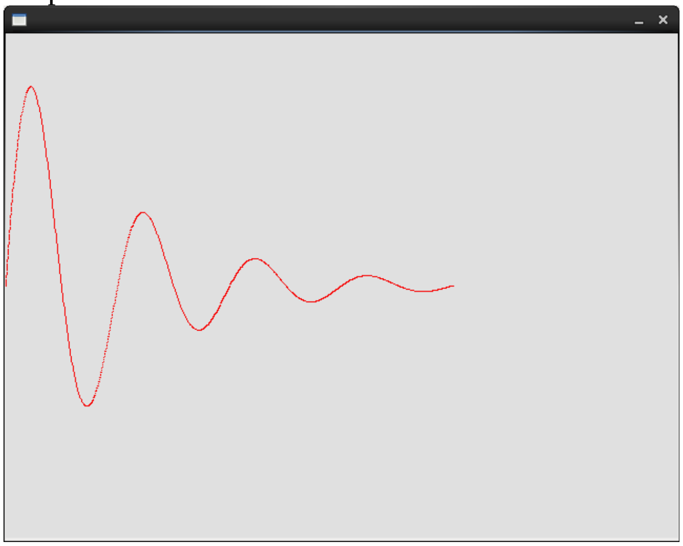
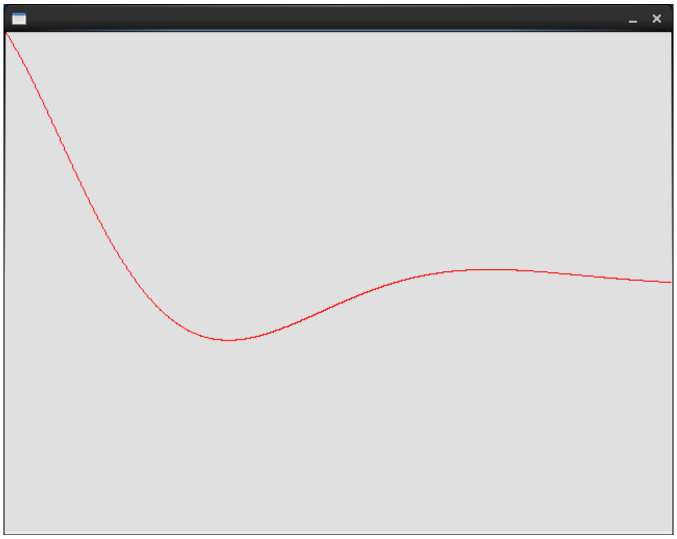
Yazhuo Liu
Lab 3
Exercise 1
Output:



$f(x) = e^{-|x|} \sin (2\pi x)$, VWIDTH/6.0



$f(x) = e^{-|x|} \sin (2x)$, VWIDTH/4.0

Partial Code:
//plots.cpp

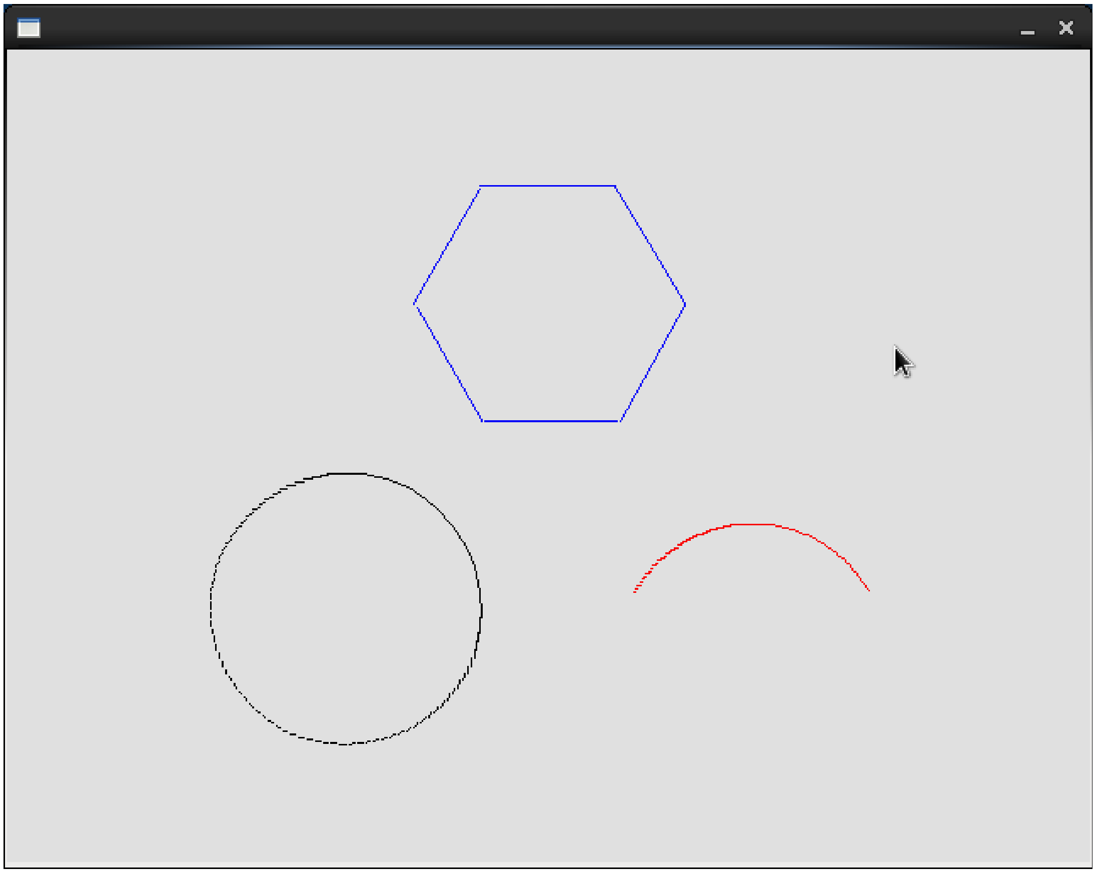
```
...
double f ( double x )
{
  double y = exp ( -fabs ( x ) ) * sin ( 2 * 3.1415926 * x );
  //double y = exp ( -fabs ( x ) ) * sin ( 2 * x );

  return y;
}
...
double x, y, a, b, c, d;
//a = VWIDTH / 4.0;
a = VWIDTH / 6.0;
c = 0;
...
```

Report:

I changed the function to $f(x) = e^{-|x|} \sin (2\pi x)$ and $f(x) = e^{-|x|} \sin (2 x)$, and I change the scaling to VWIDTH/6.0, and the graphs changed accordingly.

Exercise 2

Output:



Partial Code:

```
...
//draw a polygon
void draw_polygon ( Surface &surf, int n,  int radius, float rotAngle )
{
  if ( n < 3 ) return;                                   //bad number of sides
  int cx = surf.getCP().x;
  int cy = surf.getCP().y;
  double angle = rotAngle * 3.14159265 / 180;         //initial angle
  double angleInc = 2 * 3.14159265 / n;               //angle increment
  surf.moveTo ( ( int) (radius * cos( angle ) + cx),
                         ( int ) ( radius * sin ( angle ) + cy ) );
  for ( int k = 0; k < n; k++ ) {                      //repeat n times
    angle += angleInc;
    surf.lineTo ( ( int) (radius * cos( angle ) + cx),
                         ( int ) ( radius * sin ( angle ) + cy ) );
  }
}

//draw an arc
```

```
void draw_arc ( Surface &surf, int n,  int radius, float rotAngle, float sweep )
{
  if ( n < 3 ) return;                                    //bad number of sides
  int cx = surf.getCP().x;
  int cy = surf.getCP().y;
  double angle = rotAngle * 3.14159265 / 180;         //initial angle
  double angleInc = sweep * 3.14159265 / (180 * n);           //angle increment
  surf.moveTo ( ( int) (radius * cos( angle ) + cx),
                        ( int ) ( radius * sin ( angle ) + cy ) );
  for ( int k = 0; k < n; k++ ) {                        //repeat n times
    angle += angleInc;
    surf.lineTo ( ( int) (radius * cos( angle ) + cx),
                        ( int ) ( radius * sin ( angle ) + cy ) );
  }
}
...
int main()
{
...
  //draw an hexagon
  surf.setColor ( 0, 0, 0xff );               //using blue color
  surf.moveTo ( center.x, center.y - 90  );
  draw_polygon ( surf, 6, 80, 0 );

  //draw a circle
  surf.setColor (0, 0, 0);
  surf.moveTo (center.x - 120, center.y + 90);
  draw_polygon (surf, 100, 80, 0);

  //draw an arc
  surf.setColor (0xff, 0, 0);
  surf.moveTo (center.x + 120, center.y + 120);
  draw_arc ( surf, 80, 80, 210, 120 );
...
}
```

Report:

The hexagon and circle was relatively easy, but the arc was a little bit tricky. I looked the sample code and made some modifications in the original code, I changed the angle increment so it will become an arc instead of a circle. I also learned how the start angle works. I feel I did well on this lab.