ST. XAVIER’S COLLEGE

**(Affiliated to Tribhuvan University)**

Maitighar, Kathmandu



**CG Lab Assignment #7**

**Translation, Rotation and Scaling**

**Submitted by:**

Divya Jyoti Pokharel

013BSCCSIT019

**Submitted to:**

|  |  |
| --- | --- |
| Er. Anil Sah  Lecturer Department of Computer Science |  |

**Date of Submission: September 1, 2015**

**Statement:**

Program for translation, rotation and scaling.

**Source code:**

//---------------------------------------------------------------------------

#include <vcl\vcl.h>

#pragma hdrstop

#include <math.h>

#include "rotat.h"

//---------------------------------------------------------------------------

#pragma resource "\*.dfm"

TForm1 \*Form1;

//---------------------------------------------------------------------------

\_\_fastcall TForm1::TForm1(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::translateClick(TObject \*Sender)

{

int tx,ty,i,j;

int h=input->Height;

int w=input->Width;

tx=StrToInt(inX->Text);

ty=StrToInt(inY->Text);

int a,b;

for(i=0;i<=h;i++){

for(j=0;j<=w;j++){

a=i+tx;

b=j+ty;

output->Canvas->Pixels[a][b] = input->Canvas->Pixels[i][j];

}

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::rotateClick(TObject \*Sender)

{

int angle,i,j;

float radian;

int h=input->Height;

int w=input->Width;

angle=StrToInt(inangle->Text);

radian=(angle\*3.14)/180;

int a,b;

for(i=0;i<=h;i++){

for(j=0;j<=w;j++){

a=i\*cos(radian)-j\*sin(radian);

b=i\*sin(radian)+j\*cos(radian);

output->Canvas->Pixels[a][b] = input->Canvas->Pixels[i][j];

}

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::scaleClick(TObject \*Sender)

{

int i,j;

float sx,sy;

float radian;

int h=input->Height;

int w=input->Width;

sx=StrToFloat(inSx->Text);

sy=StrToFloat(inSy->Text);

int a,b;

for(i=0;i<=h;i++){

for(j=0;j<=w;j++){

a=i\*sx;

b=j\*sy;

output->Canvas->Pixels[a][b] = input->Canvas->Pixels[i][j];

}

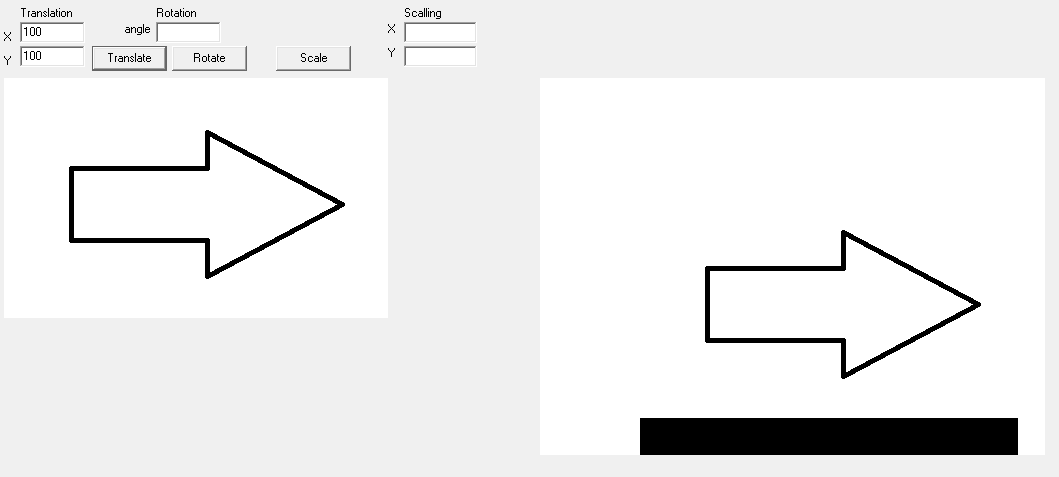
}

}

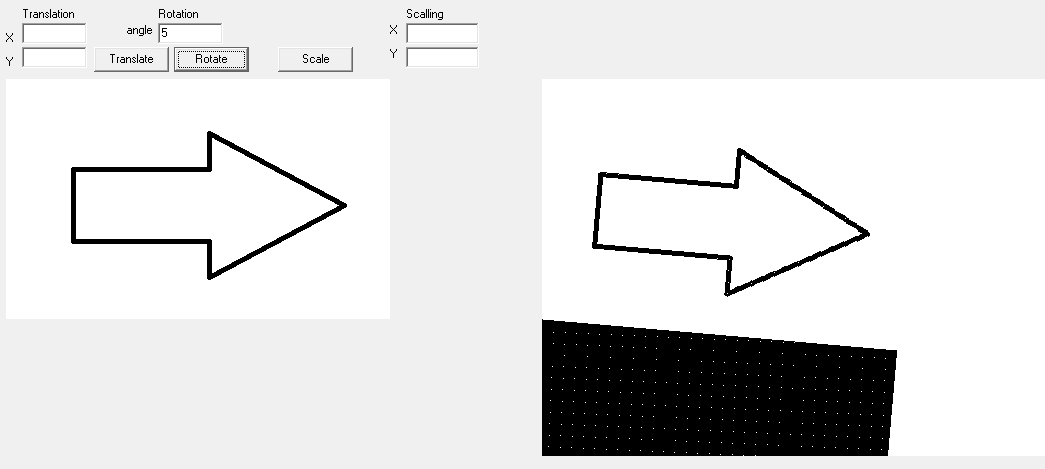
//---------------------------------------------------------------------------

**Output**

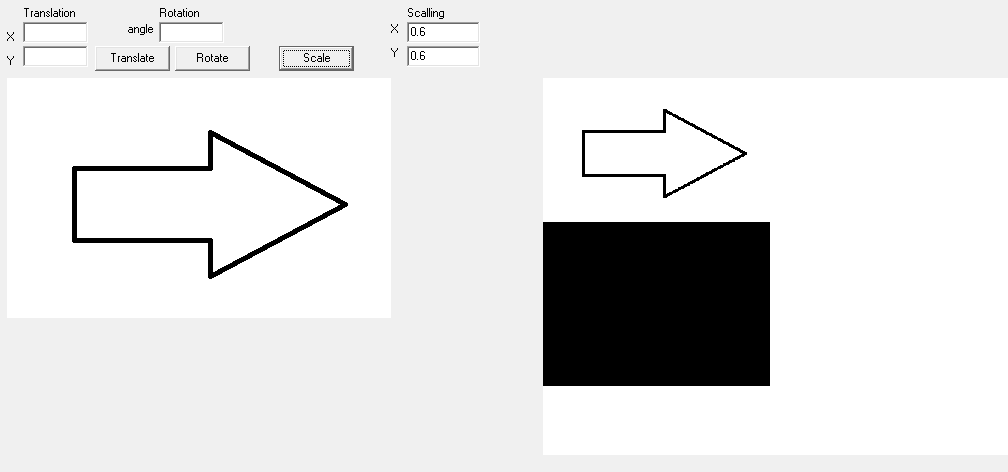
**Translation**

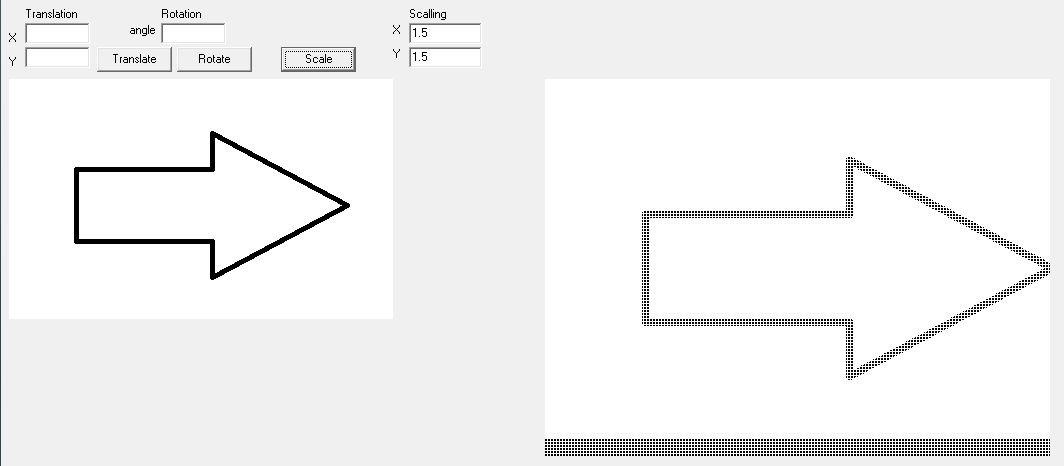


**Rotation**



**Scaling**





**Conclusion:**

Hence, the translation, rotation and calling was performed using C++ builder.