**Assignment no: 03**

**//Write a C++ class for a Line drawing method using overloading DDA and Bresenham’s Algorithms, inheriting the pixel or point.**

#include<graphics.h>

#include<iostream>

using namespace std;

class pt //base class

{

protected: int xco,yco,color;

public:

pt()

{

xco=0;yco=0;color=15;

}

void setco(int x,int y)

{

xco=x;

yco=y;

}

void setcolor(int c)

{

color=c;

}

void draw()

{ putpixel(xco,yco,color);

}

};

class dline: public pt //derived class

{

private: int x2,y2;

public:

dline():pt()

{

x2=0,y2=0;

}

void setline(int x, int y, int xx, int yy)

{

pt::setco(x,y);

x2=xx;

y2=yy;

}

void drawl() //Bresenham's Line

{

float x,y,dx,dy,e,temp;

int i,s1,s2,ex;

dx=abs(x2-xco);

dy=abs(y2-yco);

x=xco;

y=yco;

pt::setco(x,y);

pt::draw();

if(x2 > xco) //sign() function

{

s1=1;

}

if(x2 < xco)

{

s1=-1;

}

if(x2 == xco)

{

s1=0;

}

if(y2 > yco)

{

s2=1;

}

if(y2 < yco)

{

s2=-1;

}

if(y2 == yco)

{

s2=0;

}

if(dy > dx)

{

temp = dx;

dx = dy;

dy = temp;

ex = 1;

}

else

{

ex=0;

}

e=2\*dy-dx; //decision variable

i=1;

do

{

while(e>=0)

{

if(ex==1)

{

x = x + s1;

}

else

{

y = y + s2;

}

e = e + 2\*dy - 2\*dx;

}

if(ex==1)

{

y = y + s2;

}

else

{

x = x + s1;

}

e = e + 2\*dy;

pt::setco(x,y);

pt::draw();

i = i + 1;

}while(i<=dx);

}

void drawl(int colour) //DDA Line

{

float x,y,dx,dy,len;

int i;

pt::setcolor(colour);

dx=abs(x2-xco);

dy=abs(y2-yco);

if(dx >= dy)

{

len=dx;

}

else

{

len=dy;

}

dx=(x2-xco)/len;

dy=(y2-yco)/len;

x = xco + 0.5;

y = yco + 0.5;

i=1;

while(i<=len)

{

pt::setco(x,y);

pt::draw();

x = x + dx;

y = y + dy;

i = i + 1;

cout<<"\ti"<<i;

cout<<"\tx"<<x;

cout<<"\ty "<<y<<endl;

}

pt::setco(x,y);

pt::draw();

}

};

int main()

{

int gd=DETECT,gm=VGAMAX;

int ch,x1,y1,x2,y2, xmax,ymax,xmid,ymid;

char a;

initgraph(&gd,&gm,NULL);

pt p;

dline dda;

xmax = getmaxx();

ymax = getmaxy();

xmid = xmax /2;

ymid = ymax /2;

line(xmid,0,xmid,ymax); //Y co-ordinate

line(0,ymid,xmax,ymid); //X co-ordinate

do

{

xmax = getmaxx();

ymax = getmaxy();

xmid = xmax /2;

ymid = ymax /2;

cout<<"1.DDA LINE..";

cout<<"\n2.BRESENHAM'S LINE..";

cout<<"\n3.EXIT..";

cout<<"\nEnter your choice: ";

cin>>ch;

switch(ch)

{

case 1:

cout<<"\n Enter x1: "; cin>>x1;

cout<<"\n Enter y1: "; cin>>y1;

cout<<"\n Enter x2: "; cin>>x2;

cout<<"\n Enter y2: "; cin>>y2;

dda.setline(x1+xmid,ymid-y1,x2+xmid,ymid-y2);

dda.drawl(15);

break;

case 2:

cout<<"\n Enter x1: "; cin>>x1;

cout<<"\n Enter y1: "; cin>>y1;

cout<<"\n Enter x2: "; cin>>x2;

cout<<"\n Enter y2: "; cin>>y2;

dda.setline(x1+xmid,ymid-y1,x2+xmid,ymid-y2);

dda.drawl();

break;

case 3:

exit;

break;

}

cout<<"\nDO U Want To Continue y OR n: ";

cin>>a;

}while(a!='n');

delay(3000);

getch();

closegraph();

return 0;

}