Write C++/Java program for line drawing using DDA or Bresenham’s algorithm with

patterns such as solid, dotted, dashed, dash dot and thick

[LineStyles](https://abhiyatrana.files.wordpress.com/2017/03/linestyles.odt)

//\*\*\*\*\*\*\*\* linestyles.h\*\*\*\*\*\*\*\*

#ifndef LINESTYLES\_H

#define LINESTYLES\_H

#include

class LineStyles : public QWidget

{

Q\_OBJECT

public:

LineStyles(QWidget \*parent = 0);

~*LineStyles*();

void *paintEvent*(QPaintEvent \*e);

void DDALineThick(QPainter \*qp,int x1p,int y1p,int x2p,int y2p);

void DDALineDash(QPainter \*qp,int x1p,int y1p,int x2p,int y2p);

void DDALineDotted(QPainter \*qp,int x1p,int y1p,int x2p,int y2p);

void DDALineDashDotted(QPainter \*qp,int x1p,int y1p,int x2p,int y2p);

// void Bresenhams(QPainter \*qp);

};

#endif // LINESTYLES\_H

//\*\*\*\*\* linestyles.cpp\*\*\*\*\*\*

#include "linestyles.h"

#include

LineStyles::LineStyles(QWidget \*parent)

: QWidget(parent)

{

}

LineStyles::~*LineStyles*()

{

}

void LineStyles::*paintEvent*(QPaintEvent \*e)

{

Q\_UNUSED(e);

QPainter qp(this);

DDALineThick(&qp,50,50,200,50);

DDALineThick(&qp,50,52,200,52);

DDALineThick(&qp,50,54,200,54);

DDALineDash(&qp,50,70,200,70);

DDALineDotted(&qp,50,90,200,90);

DDALineDashDotted(&qp,50,110,200,110);

}

void LineStyles::DDALineThick(QPainter \*qp,int x1p,int y1p,int x2p,int y2p)

{

QPen p(Qt::red,2,Qt::SolidLine);

qp->setPen(p);

int x1=x1p;

int y1=y1p;

int x2=x2p;

int y2=y2p;

int dx,dy,length,i;

float xincr,yincr,x,y;

dx=abs(x2-x1);

dy=abs(y2-y1);

if(dx>=dy)

length=dx;

else

length=dy;

xincr=(x2-x1)/length;

yincr=(y2-y1)/length;

x=x1+0.5;

y=y1+0.5;

qp->drawPoint((int)x,(int)y);

i=1;

while(i<=length) { x=x+xincr; y=y+yincr; qp->drawPoint((int)x,(int)y);

i=i+1;

}

}

void LineStyles::DDALineDash(QPainter \*qp,int x1p,int y1p,int x2p,int y2p)

{

QPen p(Qt::red,2,Qt::SolidLine);

qp->setPen(p);

int x1=x1p;

int y1=y1p;

int x2=x2p;

int y2=y2p;

int i;

float dx,dy,length;

float xincr,yincr,x,y;

dx=abs(x2-x1);

dy=abs(y2-y1);

if(dx>=dy)

length=dx;

else

length=dy;

xincr=(x2-x1)/length;

yincr=(y2-y1)/length;

x=x1+0.5;

y=y1+0.5;

qp->drawPoint((int)x,(int)y);

i=1;

while(i<=length && x<=x2)

{

x=x+xincr;

y=y+yincr;

if(i%11==0)

{

x=x+5;

}

else

{

qp->drawPoint((int)x,(int)y);

}

i=i+1;

}

}

void LineStyles::DDALineDashDotted(QPainter \*qp,int x1p,int y1p,int x2p,int y2p)

{

QPen p(Qt::red,2,Qt::SolidLine);

qp->setPen(p);

int x1=x1p;

int y1=y1p;

int x2=x2p;

int y2=y2p;

int i;

float dx,dy,length;

float xincr,yincr,x,y;

dx=abs(x2-x1);

dy=abs(y2-y1);

if(dx>=dy)

length=dx;

else

length=dy;

xincr=(x2-x1)/length;

yincr=(y2-y1)/length;

x=x1+0.5;

y=y1+0.5;

qp->drawPoint((int)x,(int)y);

i=1;

while(i<=length && x<=x2)

{

x=x+xincr;

y=y+yincr;

if(i%11==0)

{

x=x+8;

qp->drawPoint((int)x-4,(int)y);

}

else

{

qp->drawPoint((int)x,(int)y);

}

i=i+1;

}

}

void LineStyles::DDALineDotted(QPainter \*qp,int x1p,int y1p,int x2p,int y2p)

{

QPen p(Qt::red,2,Qt::SolidLine);

qp->setPen(p);

int x1=x1p;

int y1=y1p;

int x2=x2p;

int y2=y2p;

int dx,dy,length,i;

float xincr,yincr,x,y;

dx=abs(x2-x1);

dy=abs(y2-y1);

if(dx>=dy)

length=dx;

else

length=dy;

xincr=(x2-x1)/length;

yincr=(y2-y1)/length;

x=x1+0.5;

y=y1+0.5;

qp->drawPoint((int)x,(int)y);

i=1;

while(i<=length)

{

x=x+xincr;

y=y+yincr;

if(i%7==0)

{

qp->drawPoint((int)x,(int)y);

}

i=i+1;

}

}

//\*\*\*\*main.cpp\*\*\*\*\*\*\*\*\*\*\*

#include "linestyles.h"

#include

int main(int argc, char \*argv[])

{

QApplication a(argc, argv);

LineStyles w;

w.show();

return a.exec();

}

//\*\*\*\*\*Output\*\*\*\*\*\*\*\*\*