```
//Program: DDA and Bresenham's Line Drawing Algorithm using pixel class and
function overloading
#include<iostream>
#include<graphics.h>
#include<math.h>
using namespace std;

int sign(int arg)
  {
   if(arg<0)
     return -1;
   if(arg==0)
     return 0;
   if(arg>0)
     return 1;
  }

//BASE CLASS START
class base
{
 public:
  void linealgo(int x1, int y1, int x2, int y2)    //bresenham's algo
  {
   int x,y,xx=(x2-x1), yy=(y2-y1);
   int dx = abs(xx), dy = abs(yy);

   //interchange x & y depending on the slope of the line
   if(dy>dx)
   {
    int temp = x, temp1=dx;
    x=y;
    dx=dy;
    y=temp;
    dy=temp1;    //steep slope
   }
   else// else gentle slope
    cout<<"\n\t|dx|>=|dy| HENCE gentle slope: ";

   int s1,s2,exchange;
   s1= sign(xx);
   s2= sign(yy);

   int g = ((2*dy)-dx);

   x=x1; y=y1;
   int i=1;

   putpixel(x1,y1, WHITE);

   while(i<=dx)
   {
    if(g>=0)
     {
      x = x + s1;
      y = y + s2;
      g = (g+2*dy-2*dx);
     }
    else
     {
      x = x + s1;
      g = (g+2*dy);
     }
    putpixel(x,y, WHITE);
    i++;
   };
  }
```

```
};//BASE CLASS END

class pixel : public base  //CHILD CLASS START
{
 public:


  void linealgo(float x1, float y1, float x2, float y2)  //DDA Algo
  {
   float xnew, ynew;
   int steps, dx=(x2-x1), dy=(y2-y1);

   if(abs(dx)>abs(dy))
     steps=abs(dx);
   else
     steps=abs(dy);

   float xinc=(float)dx/steps, yinc=(float)dy/steps;
   float x=x1, y=y1;

   putpixel(x,y,WHITE); //putpixel(x,y,color) // Also we can write as
putpixel(x1,y1, WHITE)

   int a,b;

   for(int i=1 ; i<=steps ; i++)
   {
    x = (x + xinc);
    y = (y + yinc);


    //FOR CONVERTING THE FLOATING VALUE TO ITS NEAREST INTEGER VALUE i.e. same as
use of floor or ceil f(x)
    a=x + 0.5;
    b=y + 0.5;

    putpixel(a,b,WHITE);
   }
  }
}obj;       //object of class pixel

int main()
{
 int ch;

 cout<<"\n Enter the choice[1.DDA algo  2.Bresenham's algo]: "<<endl;
 cin>>ch;

 switch(ch)
 {
  case 1:{//FOR DDA ALGO
          float x1,x2,y1,y2;

          cout<<"\n Enter the coordinates (x1,y1,x2,y2): "<<endl;
          cin>>x1>>y1>>x2>>y2;

          int gd=DETECT, gm;
          initgraph(&gd,&gm, NULL);

          obj.linealgo(x1,y1,x2,y2);       //function overloading

          getch();
          closegraph();

          break;
         }
```

```
  case 2:{//FOR BRESENHAM'S ALGO
          int x1,x2,y1,y2;

          cout<<"\n Enter the coordinates (x1,y1,x2,y2): "<<endl;
          cin>>x1>>y1>>x2>>y2;

          int gd=DETECT, gm;
          initgraph(&gd,&gm, NULL);

          obj.linealgo(x1,y1,x2,y2);        //function overloading

          getch();
          closegraph();

          break;
         }
 }

 return(0);
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
                                                OUTPUT
gauravgarje@gaurav-Inspiron-3542:~$ g++ cgprac4.cpp -lgraph
gauravgarje@gaurav-Inspiron-3542:~$ ./a.out

 Enter the choice[1.DDA algo  2.Bresenham's algo]:
1

 Enter the coordinates (x1,y1,x2,y2):
50
50
350
350
gauravgarje@gaurav-Inspiron-3542:~$ ./a.out

 Enter the choice[1.DDA algo  2.Bresenham's algo]:
2

 Enter the coordinates (x1,y1,x2,y2):
50
50
150
300
gauravgarje@gaurav-Inspiron-3542:~$
```