<-->

Sample Source Code :-

```
#include<iostream>
#include<graphics.h>
#include<stdlib.h>

using namespace std;

class point
{
   public:
   int x,y;
};

class poly
{
   private:
   point p[20];
   int inter[20],x,y;
   int v,xmin,ymin,xmax,ymax;
   public:
   int c;
   void read();
   void calcs();
   void display();
   void ints(float);
   void sort(int);
};

void poly::read()
{
   int i;
   cout << "\n Scan Fill Alogrithm ";
   cout << "\n Enter Number of vertices of polygon: ";
   cin >> v;
   if (v>2)
   {
      for (i = 0; i < v; i++) //Accept the values
      {
         cout << "\n Enter the co-ordinate no. " << i+1 << " : ";
         cout << "\n\tx" << i+1 << "=";
         cin >> p[i].x;
         cout << "\n\ty" << i+1 << "=";
         cin >> p[i].y;
      }

      p[i].x = p[0].x;
      p[i].y = p[0].y;
      xmin = xmax = p[0].x;
```

```
      ymin = ymax = p[0].y;
   }

   else
   cout << "\n Enter valid no. of vertices.";
}

void poly::calcs()
{
   for (int i=0; i < v ;i++)
   {
      if(xmin>p[i].x)
      xmin=p[i].x;
      if(xmax<p[i].x)
      xmax=p[i].x;
      if(ymin>p[i].x)
      ymin=p[i].x;
      if(ymax<p[i].x)
      ymax=p[i].x;
   }
}

void poly::display()
{
   int ch1;
   char ch='y';
   float s ,s2;
   do
   {
      cout << "\nMenu:";
      cout << "\n\n\t1 . Scan line fill";
      cout << "\n\n\t2 . Exit ";
      cout << "\nEnter your choice:";
      cin >> ch1;
      switch(ch1)
      {
         case 1:
         s = ymin + 0.01;
         delay(100);
         cleardevice();
         while (s<=ymax)
         {
            ints(s);
            sort(s);
            s++;
         }

         break;
         case2:
         exit(0);
      }
```

```
      cout << "Do you want to continue? : ";
      cin >> ch;

   }
   while (ch=='y' || ch=='Y');
}

void poly::ints(float z)
{
   int x1,x2,y1,y2,temp;
   c=0;
   for (int i=0; i<v;i++)
   {
      x1 = p[i].x;
      y1 = p[i].y;
      x2 = p[i+1].x;
      y2 = p[i+1].y;
      if (y2<y1)
      {
         temp = x1;
         x1=x2;
         x2=temp;
         temp = y1;
         y1=y2;
         y2=temp;
      }
      if (z<=y2 && z>=y1)
      {
         if((y1-y2)==0)
         x=x1;
         else
         {
            x= ((x2-x1)*(z-y1))/(y2-y1);
            x=x+x1;
         }
         if(x<=xmax && x>=xmin)
         inter[c++]=x;
      }
   }
}

void poly::sort(int z) //sorting
{
   int temp,j,i;
   for (i=0; i<v;i++)
   {
      line(p[i].x,p[i].y,p[i+1].x,p[i+1].y);
   }
   delay(50);
   for (i=0;i<c;i+=2)
   {
```

```
      delay(50);
      line(inter[i],z,inter[i+1],z);
   }
}

int main() //main
{
   int cl;
   int gd = DETECT, gm;
   initgraph(&gd,&gm,"C:\\tc\\bgi");
   cleardevice();
   poly x;
   x.read();
   x.calcs();
   cleardevice();
   cout << "\n\t Enter the colour you want :(in range 0 to 15) ->"; //selcting color
   cin >> cl;
   setcolor(cl);
   x.display();

   closegraph(); //closing graph
   getch();
   return 0;
}
```

<--->

Sample Output :-