Name: Khushi Nitinkumar Patel

PRN: 2020BTECS00037

Batch: T5

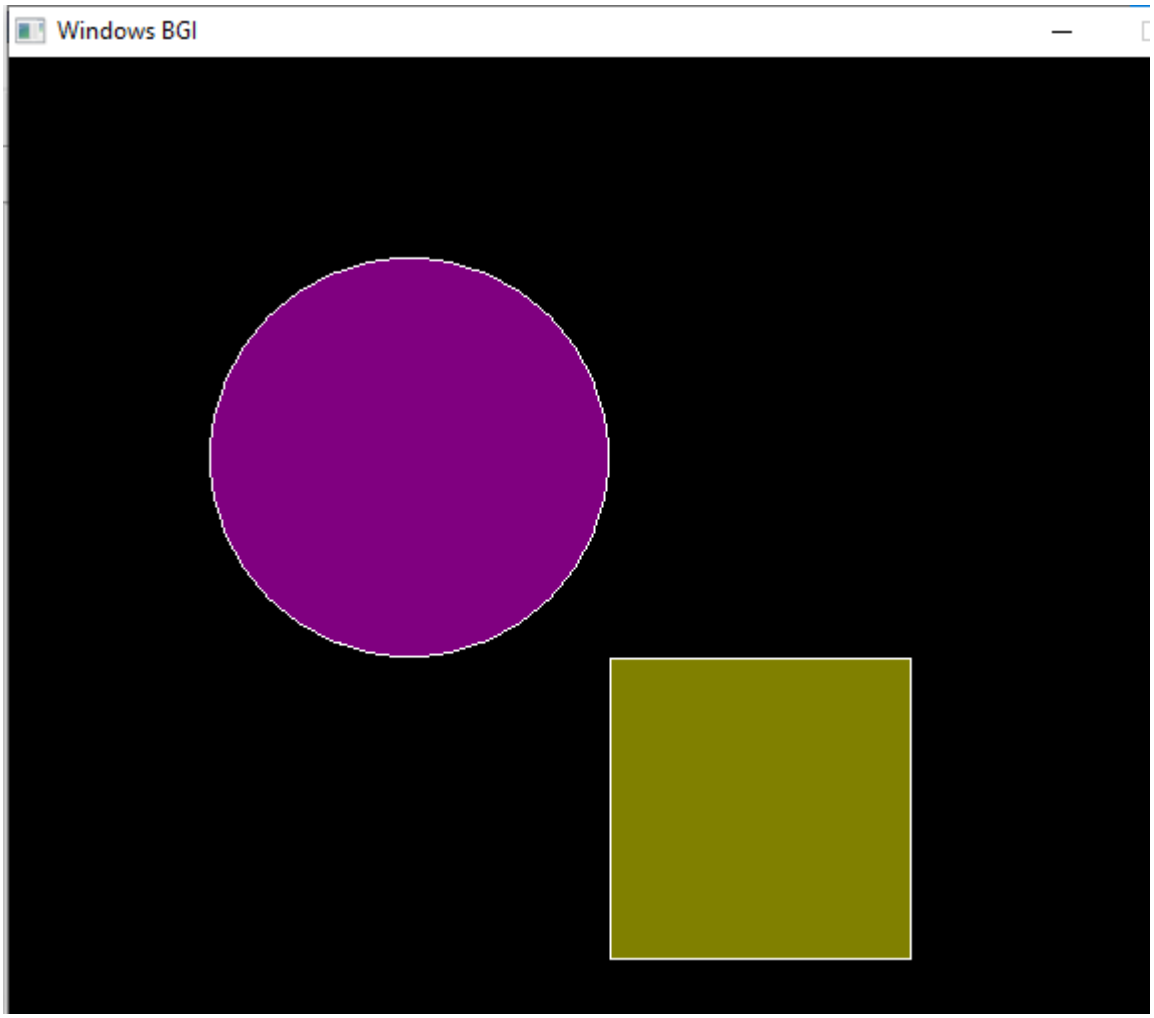
Experiment 8: Implementation of Object filling algorithm

Code:

cg_7.cpp

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
#include<process.h>
#include<graphics.h>

void boundaryFill4(int x, int y, int fill_color, int boundary_color)
{
if (getpixel(x, y) != boundary_color && getpixel(x, y) != fill_color)
{
putpixel(x, y, fill_color);
boundaryFill4(x + 1, y, fill_color, boundary_color);
boundaryFill4(x, y + 1, fill_color, boundary_color);
boundaryFill4(x - 1, y, fill_color, boundary_color);
boundaryFill4(x, y - 1, fill_color, boundary_color);
}
}

void boundaryFill8(int x, int y, int fill_color, int boundary_color)
{
if (getpixel(x, y) != boundary_color && getpixel(x, y) != fill_color)
{
putpixel(x, y, fill_color);
boundaryFill8(x + 1, y, fill_color, boundary_color);
boundaryFill8(x, y + 1, fill_color, boundary_color);
boundaryFill8(x - 1, y, fill_color, boundary_color);

boundaryFill8(x, y - 1, fill_color, boundary_color);
boundaryFill8(x - 1, y - 1, fill_color, boundary_color);
boundaryFill8(x - 1, y + 1, fill_color, boundary_color);
boundaryFill8(x + 1, y - 1, fill_color, boundary_color);
boundaryFill8(x + 1, y + 1, fill_color, boundary_color);
}
}
int main()
{
int gd = DETECT, gm;
initgraph(&gd, &gm, "c:\\turboc3\\bgi");
circle(200, 200, 100);
rectangle(300,300,450,450);
boundaryFill4(200,200,5,15);
boundaryFill8(355, 355, 6, 15);
getch();
closegraph();
return 0;
}

```

Output:



- **Flood Fill**

Code:

cg_7.cpp [*] Untitled2

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <process.h>
#include <graphics.h>

void floodFill(int x, int y, int fill_color, int old_color)
{
if (getpixel(x, y) == old_color)
{

putpixel(x, y, fill_color);

floodFill(x + 1, y, fill_color, old_color);
floodFill(x, y + 1, fill_color, old_color);
floodFill(x - 1, y, fill_color, old_color);
floodFill(x, y - 1, fill_color, old_color);
}
}

int main()
{
int gd = DETECT, gm;
initgraph(&gd, &gm, "c:\\turboc3\\bgi");

circle(200, 200, 100); floodFill(200,200,14,0); getch();
closegraph(); return 0;
}
```

Output:

- **Scan-line Filling**

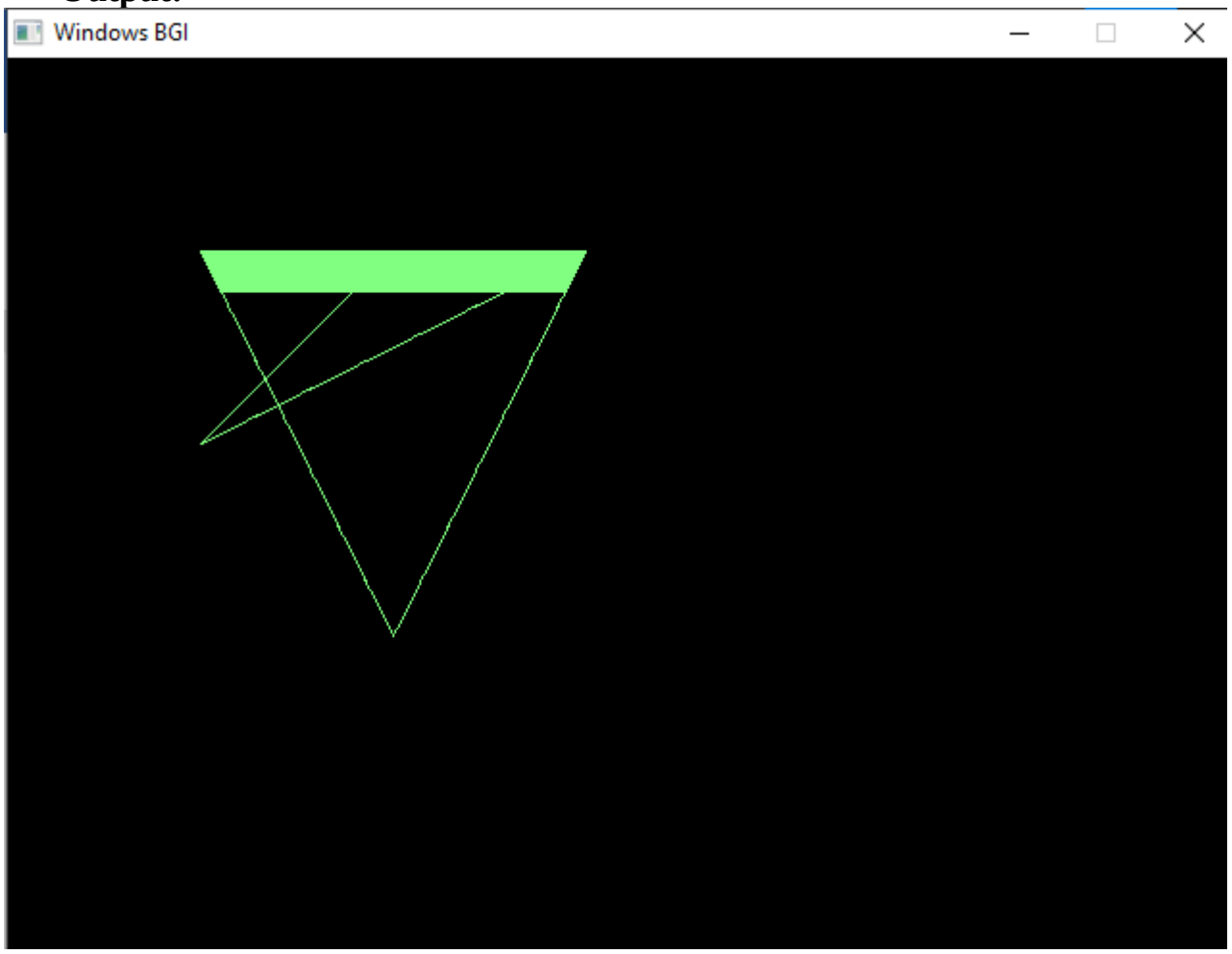
Code:

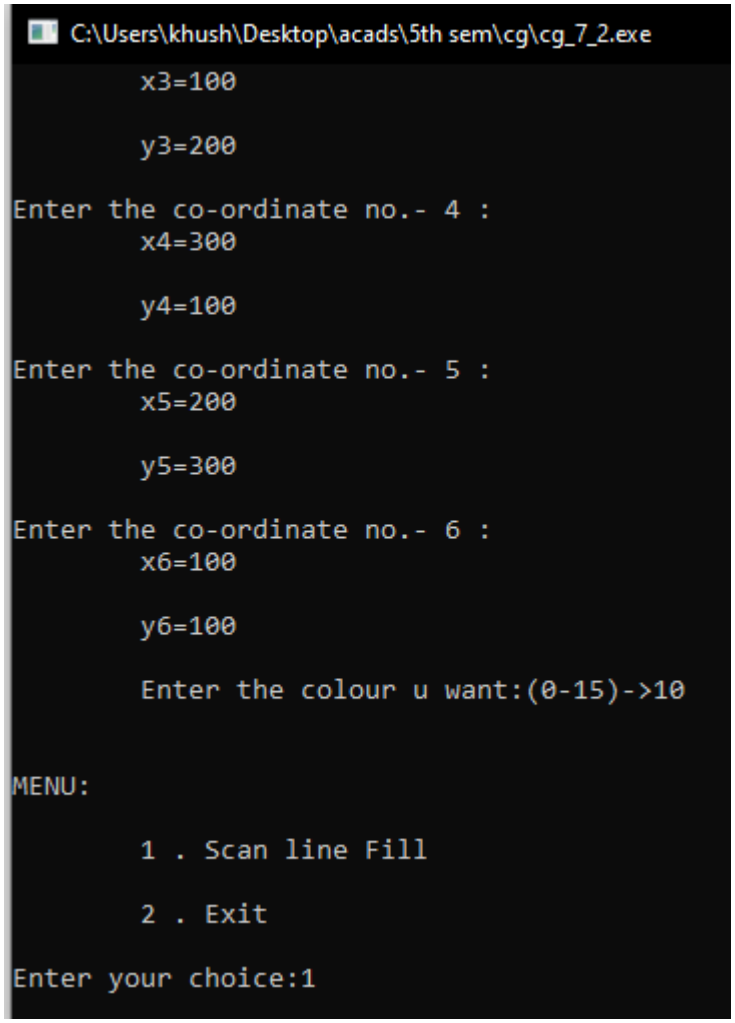
```
#include <conio.h>
#include <iostream>
#include <graphics.h>
#include <stdlib.h>
using namespace std;


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\t SCAN_FILL ALGORITHM";
cout<<"\n Enter the no of vertices of polygon:"; cin>>v;
if(v>2)
{
for(i=0;i<v; i++)
{
cout<<"\nEnter 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].y)
ymin=p[i].y;
if(ymax<p[i].y)
ymax=p[i].y;
}

}

void poly::display()
{
int ch1; char ch='y'; float s,s2; do
{
cout<<"\n\nMENU:";
cout<<"\n\n\t1 . Scan line Fill ";
cout<<"\n\n\t2 . Exit ";
cout<<"\n\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; case 2:
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)
{
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(100); for(i=0; i<c;i+=2)
{
delay(50); line(inter[i],z,inter[i+1],z);
}
}

int main()
{
int cl;
int gd = DETECT, gm;
initgraph(&gd, &gm, "c:\\turboc3\\bgi");
cleardevice();
poly x; x.read();

x.calcs(); cleardevice();
cout<<"\n\tEnter the colour u want:(0-15)->";
cin>>cl;
setcolor(cl);
x.display(); c
losegraph();
getch();
return 0;
}
```

Output:




C:\Users\khush\Desktop\acads\5th sem\cg\cg_7_2.exe
x3=100
y3=200
Enter the co-ordinate no.- 4 :
x4=300
y4=100
Enter the co-ordinate no.- 5 :
x5=200
y5=300
Enter the co-ordinate no.- 6 :
x6=100
y6=100
Enter the colour u want:(0-15)->10
MENU:
1 . Scan line Fill
2 . Exit
Enter your choice:1