

# Computer Graphics Lab

## (B. Sc. 6<sup>th</sup> Semester)

### 1. Fundamental Graphics Functions

```
#include<stdio.h>
#include<conio.h>
#include<process.h>
#include<graphics.h>
void main()
{
int gd=DETECT,gm,ch;
initgraph(&gd,&gm," ");
do
{
clrscr();
printf("\nmenu\n1.circle\n2.line\n3.rectangle\n4.exit\nEnter your choice:
");
scanf("%d",&ch);
cleardevice();
outtextxy(10,10,"FUNDAMENTALS");
switch(ch)
{
case 1:
circle(200,200,80);
break;
case 2:
line(200,200,300,300);
break;
case 3:
rectangle(100,100,400,400);
break;
}
getch();
}while(ch<=3);
getch();
}
```

\*\*\*\*\*

## 2. DDA Line Drawing Algorithm Using C Programming

```
#include<stdio.h>
#include<math.h>
#include<conio.h>
#include<graphics.h>
#define round(val) (int)(val+0.5)
void main()
{
int gd=DETECT, gm;
void line_dda(int, int, int, int);
int xa, xb, ya, yb;
printf("Enter the two values");
scanf("%d%d%d%d", &xa, &ya, &xb, &yb);  initgraph(&gd, &gm, "");
cleardevice();
line_dda(xa, ya, xb, yb);
getch();
closegraph();
}

void line_dda(int xa, int ya, int xb, int yb)
{
int Dx=xb-xa, Dy=yb-ya, steps, k;
float xin, yin, X=xa, Y=ya;
if (abs(Dx)>abs(Dy))
    steps=abs(Dx);
else
    steps=abs(Dy);

xin=Dx/(float) steps;
yin=Dy/(float) steps;
putpixel(round(X), round(Y), 6);

for (k=0; k<steps; k++)
{
X=X+xin;
Y=Y+yin;
putpixel(round(X), round(Y), 6);
}
}
```

\*\*\*\*\*

### 3. Bresenham's Line Drawing Algorithm

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

void main()
{
int x1,x2,y1,y2;
int gd=DETECT,gm;
void linebres(int,int,int,int);
printf("Enter the two end points:");
scanf("%d%d%d%d",&x1,&x2,&y1,&y2);
initgraph(&gd,&gm,"");
cleardevice();
linebres(x1,y1,x2,y2);
getch();
line(x1,y1,x2,y2);
getch();
closegraph();
}

void linebres(int x1,int y1,int x2,int y2)
{
int dx=abs(x1-x2),dy=abs(y1-y2);
int p,x,y,i,xend,yend;
if(dx!=0)
{
p=2*dy-dx;
if(x1>x2)
{
x=x2;
y=y2;
xend=x1;
}
else
{
x=x1;
y=y1;
xend=x2;
}
putpixel(x,y,2);
for(i=x;i<xend;i++)
{
x+=1;
if(p<0)
p+=2*dy;
else
p+=2*(dy-dx);
}
putpixel(x,y,2);
}
```

```
}
else
{
p=2*dx-dy;
if (y1>y2)
{
x=x2;
y=y2;
yend=y2;
}
putpixel(x,y,2);
for (i=y;i<yend;i++)
{
y+=1;
if (p<0)
p+=2*dx;
else
{
x+=1;
p+=2*(dx-dy);
}
putpixel(x,y,2);
}
}
}
```

\*\*\*\*\*

## 4. Bresenham's Circle Drawing Algorithm Using C Programming

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
    int gd=DETECT,gm;
    int x,y,r;
    void cir(int,int,int);
    printf("Enter the Mid points and Radius:");
    scanf("%d%d%d",&x,&y,&r);
    initgraph(&gd,&gm,"");
    cir(x,y,r);
    getch();
    closegraph();
}

void cir(int x1,int y1,int r)
{
    int x=0,y=r,p=1-r;
    void cliplot(int,int,int,int);
    cliplot(x1,y1,x,y);
    while(x<y)
    {
        x++;
        if(p<0)
            p+=2*x+1;
        else
        {
            y--;
            p+=2*(x-y)+1;
        }
        cliplot(x1,y1,x,y);
    }
}

void cliplot(int xctr,int yctr,int x,int y)
{
    putpixel(xctr +x,yctr +y,1);
    putpixel(xctr -x,yctr +y,1);
    putpixel(xctr +x,yctr -y,1);
    putpixel(xctr -x,yctr -y,1);
    putpixel(xctr +y,yctr +x,1);
    putpixel(xctr -y,yctr +x,1);
    putpixel(xctr +y,yctr -x,1);
    putpixel(xctr -y,yctr -x,1);
    getch();
}
```

\*\*\*\*\*

## 5. 2D Scaling Program Using C Programming

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

int x1,y1,x2,y2,x3,y3,mx,my;
void draw();
void scale();

void main()
{
    int gd=DETECT,gm," ";
    int c;
    initgraph(&gd,&gm," ");
    printf("Enter the 1st point for the triangle:");
    scanf("%d%d",&x1,&y1);
    printf("Enter the 2nd point for the triangle:");
    scanf("%d%d",&x2,&y2);
    printf("Enter the 3rd point for the triangle:");
    scanf("%d%d",&x3,&y3);
    draw();
    scale();
}

void draw()
{
    line(x1,y1,x2,y2);
    line(x2,y2,x3,y3);
    line(x3,y3,x1,y1);
}

void scale()
{
    int x,y,a1,a2,a3,b1,b2,b3;
    int mx,my;
    printf("Enter the scalling coordinates");
    scanf("%d%d",&x,&y);
    mx=(x1+x2+x3)/3;
    my=(y1+y2+y3)/3;
    cleardevice();
    a1=mx+(x1-mx)*x;
    b1=my+(y1-my)*y;
    a2=mx+(x2-mx)*x;
    b2=my+(y2-my)*y;
    a3=mx+(x3-mx)*x;
    b3=my+(y3-my)*y;
    line(a1,b1,a2,b2);
    line(a2,b2,a3,b3);
    line(a3,b3,a1,b1);
    draw();
}
```

```
    getch();
}
```

\*\*\*\*\*

## 6. 2D Rotation Program Using C Programming

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

void TriAngle(int x1,int y1,int x2,int y2,int x3,int y3);
void Rotate(int x1,int y1,int x2,int y2,int x3,int y3);

void main()
{
    int gd=DETECT,gm;
    int x1,y1,x2,y2,x3,y3;
    initgraph(&gd,&gm, " ");

    printf("Enter the 1st point for the triangle:");
    scanf("%d%d",&x1,&y1);
    printf("Enter the 2nd point for the triangle:");
    scanf("%d%d",&x2,&y2);
    printf("Enter the 3rd point for the triangle:");
    scanf("%d%d",&x3,&y3);
    TriAngle(x1,y1,x2,y2,x3,y3);
    getch();
    cleardevice();
    Rotate(x1,y1,x2,y2,x3,y3);
    setcolor(1);
    TriAngle(x1,y1,x2,y2,x3,y3);
    getch();
}

void TriAngle(int x1,int y1,int x2,int y2,int x3,int y3)
{
    line(x1,y1,x2,y2);
    line(x2,y2,x3,y3);
    line(x3,y3,x1,y1);
}

void Rotate(int x1,int y1,int x2,int y2,int x3,int y3)
{
    int x,y,a1,b1,a2,b2,a3,b3,p=x2,q=y2;
    float Angle;
    printf("Enter the angle for rotation:");
    scanf("%f",&Angle);
    cleardevice();
    Angle=(Angle*3.14)/180;
    a1=p+(x1-p)*cos(Angle)-(y1-q)*sin(Angle);
    b1=q+(x1-p)*sin(Angle)+(y1-q)*cos(Angle);
```

```

    a2=p+(x2-p)*cos(Angle)-(y2-q)*sin(Angle);
    b2=q+(x2-p)*sin(Angle)+(y2-q)*cos(Angle);
    a3=p+(x3-p)*cos(Angle)-(y3-q)*sin(Angle);
    b3=q+(x3-p)*sin(Angle)+(y3-q)*cos(Angle);
    printf("Rotate");
    TriAngle(a1,b1,a2,b2,a3,b3);
}
*****

```

## 7. 2D Translation Triangle Program Using C Programming

```

#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<process.h>
#include<math.h>
int x1,y1,x2,y2,x3,y3,mx,my;

void draw();
void tri();

void main()
{
    int gd=DETECT,gm;
    int c;
    initgraph(&gd,&gm,"d:\\tc\\bgi ");
    printf("Enter the 1st point for the triangle:");
    scanf("%d%d",&x1,&y1);
    printf("Enter the 2nd point for the triangle:");
    scanf("%d%d",&x2,&y2);
    printf("Enter the 3rd point for the triangle:");
    scanf("%d%d",&x3,&y3);

    cleardevice();
    draw();
    getch();
    tri();
    getch();
}

void draw()
{
    line(x1,y1,x2,y2);
    line(x2,y2,x3,y3);
    line(x3,y3,x1,y1);
}

void tri()
{
    int x,y,a1,a2,a3,b1,b2,b3;
    printf("Enter the Translation coordinates");
    scanf("%d%d",&x,&y);
    cleardevice();
    a1=x1+x;
    b1=y1+y;
    a2=x2+x;
    b2=y2+y;
    a3=x3+x;
    b3=y3+y;
    line(a1,b1,a2,b2);
    line(a2,b2,a3,b3);
}

```



```
    line(a3,b3,a1,b1);
}
```

\*\*\*\*\*

## 8. 2D Translation Rectangle Program Using C Programming

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

void RectAngle(int x,int y,int Height,int Width);
void Translate(int x,int y,int Height,int Width);

void main()
{
    int gd=DETECT,gm;
    int x,y,Height,Width;
    initgraph(&gd,&gm," ");
    printf("Enter the First point for the Rectangle:");
    scanf("%d%d",&x,&y);
    printf("Enter the Height&Width for the Rectangle:");
    scanf("%d%d",&Height,&Width);
    RectAngle(x,y,Height,Width);
    getch();
    cleardevice();
    Translate(x,y,Height,Width);
    RectAngle(x,y,Height,Width);
    getch();
}

void RectAngle(int x,int y,int Height,int Width)
{
    line(x,y,x+Width,y);
    line(x,y,x,y+Height);
    line(x+Width,y,x+Width,y+Height);
    line(x,y+Height,x+Width,y+Height);
}

void Translate(int x,int y,int Height,int Width)
{
    int Newx,Newy,a,b;
    printf("Enter the Transaction coordinates");
    scanf("%d%d",&Newx,&Newy);
    cleardevice();
    a=x+Newx;
    b=y+Newy;
    RectAngle(a,b,Height,Width);
}
```

\*\*\*\*\*

## 9. 3D Translation Program Using C Programming

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

int x1,x2,y1,y2,mx,my,depth;
void draw();
void trans();

void main()
{
    int gd=DETECT,gm,c;
    initgraph(&gd,&gm,"d:\\tc\\bgi");
    printf("\n\t\t3D Transmission\n\n");
    printf("\nEnter 1st top value(x1,y1):");
    scanf("%d%d",&x1,&y1);
    printf("Enter right bottom value(x2,y2):");
    scanf("%d%d",&x2,&y2);
    depth=(x2-x1)/4;
    mx=(x1+x2)/2;
    my=(y1+y2)/2;
    draw();
    getch();
    cleardevice();
    trans();
    getch();
}

void draw()
{
    bar3d(x1,y1,x2,y2,depth,1);
}

void trans()
{
    int a1,a2,b1,b2,dep,x,y;
    printf("\n Enter the Ttransition Co ordinates:");
    scanf("%d%d",&x,&y);
    a1=x1+x;
    a2=x2+x;
    b1=y1+y;
    b2=y2+y;
    dep=(a2-a1)/4;
    bar3d(a1,b1,a2,b2,dep,1);
    setcolor(5);
    draw();
}
```

\*\*\*\*\*

## 10. 3D Scaling Program Using C Programming

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

int x1,x2,y1,y2,mx,my,depth;
void draw();
void scale();

void main()
{
    int gd=DETECT,gm,c;
    initgraph(&gd,&gm,"d:\\tc\\bgi");
    printf("\n\t\t3D Transformation Scalling\n\n");
    printf("\nEnter 1st top value(x1,y1):");
    scanf("%d%d",&x1,&y1);
    printf("Enter right bottom value(x2,y2):");
    scanf("%d%d",&x2,&y2);
    depth=(x2-x1)/4;
    mx=(x1+x2)/2;
    my=(y1+y2)/2;
    draw();
    getch();
    cleardevice();
    scale();
    getch();
}

void draw()
{
    bar3d(x1,y1,x2,y2,depth,1);
}

void scale()
{
    int x,y,a1,a2,b1,b2,dep;
    printf("\n\n Enter scalling co-ordinates:");
    scanf("%d%d",&x,&y);
    a1=mx+(x1-mx)*x;
    a2=mx+(x2-mx)*x;
    b1=my+(y1-my)*y;
    b2=my+(y2-my)*y;
    dep=(a2-a1)/4;
    bar3d(a1,b1,a2,b2,dep,1);
    setcolor(5);
    draw();
}
```

\*\*\*\*\*

## 11. 3D Rotation Program Using C Programming

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

int x1,x2,y1,y2,mx,my,depth;
void draw();
void rotate();

void main()
{
    int gd=DETECT, gm, c;
    initgraph(&gd, &gm, "d:\\tc\\bgi");
    printf("\n3D Transformation Rotating\n\n");
    printf("\nEnter 1st top value(x1,y1):");
    scanf("%d%d", &x1, &y1);
    printf("Enter right bottom value(x2,y2):");
    scanf("%d%d", &x2, &y2);
    depth=(x2-x1)/4;
    mx=(x1+x2)/2;
    my=(y1+y2)/2;
    draw();
    getch();
    cleardevice();
    rotate();
    getch();
}

void draw()
{
    bar3d(x1,y1,x2,y2,depth,1);
}

void rotate()
{
    float t;
    int a1,b1,a2,b2,dep;
    printf("Enter the angle to rotate=");
    scanf("%f", &t);
    t=t*(3.14/180);
    a1=mx+(x1-mx)*cos(t)-(y1-my)*sin(t);
    a2=mx+(x2-mx)*cos(t)-(y2-my)*sin(t);
    b1=my+(x1-mx)*sin(t)-(y1-my)*cos(t);
    b2=my+(x2-mx)*sin(t)-(y2-my)*cos(t);
    if(a2>a1)
        dep=(a2-a1)/4;
```

```

else
    dep=(a1-a2)/4;
bar3d(a1,b1,a2,b2,dep,1);
setcolor(5);
//draw();
}

```

\*\*\*\*\*

## 12. Cohen Sutherland Line Clipping Program Using C Programming

\*\*\*\*\*

```

#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<graphics.h>
#include<dos.h>

typedef struct coordinate
{
int x,y;
char code[4];
}PT;

void drawwindow();
void drawline(PT p1,PT p2);
PT setcode(PT p);
int visibility(PT p1,PT p2);
PT resetendpt(PT p1,PT p2);

void main()
{
int gd=DETECT,v,gm;
PT p1,p2,p3,p4,ptemp;
printf("\nEnter x1 and y1\n");
scanf("%d %d",&p1.x,&p1.y);
printf("\nEnter x2 and y2\n");
scanf("%d %d",&p2.x,&p2.y);
initgraph(&gd,&gm,"c:\\turbo3\\bgi");
drawwindow();
delay(500);
drawline(p1,p2);
delay(500);
cleardevice();
delay(500);
p1=setcode(p1);
p2=setcode(p2);

```

```

v=visibility(p1,p2);
delay(500);
switch(v)
{
case 0: drawwindow();
delay(500);
drawline(p1,p2);
break;
case 1: drawwindow();
delay(500);
break;
case 2: p3=resetendpt(p1,p2);
p4=resetendpt(p2,p1);
drawwindow();
delay(500);
drawline(p3,p4);
break;
}
delay(5000);
closegraph();
}

```

```

void drawwindow()
{
line(150,100,450,100);
line(450,100,450,350);
line(450,350,150,350);
line(150,350,150,100);
}

```

```

void drawline(PT p1,PT p2)
{
line(p1.x,p1.y,p2.x,p2.y);
}

```

```

PT setcode(PT p) //for setting the 4 bit code
{
PT ptemp;
if(p.y<100)
ptemp.code[0]='1'; //Top
else
ptemp.code[0]='0';
if(p.y>350)
ptemp.code[1]='1'; //Bottom
else
ptemp.code[1]='0';
if(p.x>450)
ptemp.code[2]='1'; //Right
else
ptemp.code[2]='0';
}

```

```

if(p.x<150)
p1temp.code[3]='1'; //Left
else
p1temp.code[3]='0';
p1temp.x=p.x;
p1temp.y=p.y;
return(p1temp);
}

```

```

int visibility(PT p1,PT p2)
{
int i,flag=0;
for(i=0;i<4;i++)
{
if((p1.code[i]!='0') || (p2.code[i]!='0'))
flag=1;
}
if(flag==0)
return(0);
for(i=0;i<4;i++)
{
if((p1.code[i]==p2.code[i] && (p1.code[i]=='1'))
flag='0';
}
if(flag==0)
return(1);
return(2);
}

```

```

PT resetendpt(PT p1,PT p2)
{
PT temp;
int x,y,i;
float m,k;
if(p1.code[3]=='1')
x=150;
if(p1.code[2]=='1')
x=450;
if((p1.code[3]=='1') || (p1.code[2]=='1'))
{
m=(float)(p2.y-p1.y)/(p2.x-p1.x);
k=(p1.y+(m*(x-p1.x)));
temp.y=k;
temp.x=x;
for(i=0;i<4;i++)
temp.code[i]=p1.code[i];
if(temp.y<=350 && temp.y>=100)
return (temp);
}
if(p1.code[0]=='1')

```

```
y=100;
if(p1.code[1]=='1')
y=350;
if((p1.code[0]=='1') || (p1.code[1]=='1'))
{
m=(float)(p2.y-p1.y)/(p2.x-p1.x);
k=(float)p1.x+(float)(y-p1.y)/m;
temp.x=k;
temp.y=y;
for(i=0;i<4;i++)
temp.code[i]=p1.code[i];
return(temp);
}
else
return(p1);
}
```