Sample Code

#include<graphics.h>

#include<conio.h>

int main()

{

int gd = DETECT, gm;

initgraph(&gd, &gm, "C:**\\**TC**\\**BGI");

  putpixel(100,100,RED);

getch();

closegraph();

return 0;

}

## C program for getdrivername

#include<graphics.h>

#include<conio.h>

main()

{

int gd = DETECT, gm;

char \*drivername;

initgraph(&gd, &gm, "C:**\\**TC**\\**BGI");

drivername = getdrivername();

outtextxy(200, 200, drivername);

getch();

closegraph();

return 0;

}

## C programming code for line

#include <graphics.h>

#include <conio.h>

main()

{

int gd = DETECT, gm;

initgraph(&gd, &gm, "C:**\\**TC**\\**BGI");

line(100, 100, 200, 200);

getch();

closegraph();

return 0;

}

Drawing Arc in C.

#include <graphics.h>

#include <conio.h>

main()

{

int gd = DETECT, gm;

initgraph(&gd, &gm, "C:**\\**TC**\\**BGI");

arc(100, 100, 0, 135, 50);

getch();

closegraph();

return 0;

}

In the above program (100,100) are coordinates of center of arc, 0 is the starting angle, 135 is the end angle and 50 specifies the radius of the arc.

Drawing Bar

Declaration :- void bar(int left, int top, int right, int bottom);

ar function is used to draw a 2-dimensional, rectangular filled in bar . Coordinates of left top and right bottom corner are required to draw the bar. Left specifies the X-coordinate of top left corner, top specifies the Y-coordinate of top left corner, right specifies the X-coordinate of right bottom corner, bottom specifies the Y-coordinate of right bottom corner. Current fill pattern and fi

ll color is used to fill the bar. To change fill pattern and fill color use [setfillstyle](http://www.programmingsimplified.com/c/graphics.h/setfillstyle" \o "setfillstyle).

#include <graphics.h>

#include <conio.h>

main()

{

int gd = DETECT, gm;

initgraph(&gd, &gm, "C:**\\**TC**\\**BGI");

bar(100, 100, 200, 200);

getch();

closegraph();

return 0;

}

**Circle function in c**

Declaration :- void circle(int x, int y, int radius);

Circle function is used to draw a circle with center (x,y) and third parameter specifies the radius of the circle. The code given below draws a circle.

#include<graphics.h>

#include<conio.h>

main()

{

int gd = DETECT, gm;

initgraph(&gd, &gm, "C:**\\**TC**\\**BGI");

circle(100, 100, 50);

getch();

closegraph();

return 0;

}

In the above program (100, 100) are coordinates of center of the circle and 50 is the radius of circle.

# Cleardevice function in c

Declaration :- void cleardevice();  
cleardevice function clears the screen in graphics mode and sets the current position to (0,0). Clearing the screen consists of filling the screen with current background color.

#include <graphics.h>

#include <conio.h>

main()

{

int gd = DETECT, gm;

initgraph(&gd, &gm, "C:**\\**TC**\\**BGI");

outtext("Press any key to clear the screen.");

getch();

cleardevice();

outtext("Press any key to exit...");

getch();

closegraph();

return 0;

}

# drawpoly function in c

Drawpoly function is used to draw polygons i.e. triangle, [rectangle](http://www.programmingsimplified.com/c/graphics.h/rectangle), pentagon, hexagon etc.

Declaration :- void drawpoly( int num, int \*polypoints );

num indicates (n+1) number of points where n is the number of vertices in a polygon, polypoints points to a sequence of (n\*2) integers . Each pair of integers gives x and y coordinates of a point on the polygon. We specify (n+1) points as first point coordinates should be equal to (n+1)th to draw a complete figure.

To understand more clearly we will draw a triangle using drawpoly, consider for example the array :-  
int points[] = { 320, 150, 420, 300, 250, 300, 320, 150};

points array contains coordinates of triangle which are (320, 150), (420, 300) and (250, 300). Note that last point(320, 150) in array is same as first. See the program below and then its output, it will further clear your understanding.

#include <graphics.h>

#include <conio.h>

main()

{

int gd=DETECT,gm,points[]={320,150,420,300,250,300,320,150};

initgraph(&gd, &gm, "C:**\\**TC**\\**BGI");

drawpoly(4, points);

getch();

closegraph();

return 0;

}

## C programming code for ellipse

#include<graphics.h>

#include<conio.h>

main()

{

int gd = DETECT, gm;

initgraph(&gd, &gm, "C:**\\**TC**\\**BGI");

ellipse(100, 100, 0, 360, 50, 25);

getch();

closegraph();

return 0;

}

## C programming code for putpixel

#include<graphics.h>

#include<conio.h>

main()

{

int gd = DETECT, gm;

initgraph(&gd, &gm, "C:**\\**TC**\\**BGI");

putpixel(25, 25, RED);

getch();

closegraph();

return 0;

# getpixel function in c

getpixel function returns the color of pixel present at location(x, y).

Declaration :- int getpixel(int x, int y);

## C program for getpixel

#include<graphics.h>

#include<conio.h>

main()

{

int gd = DETECT, gm, color;

char array[50];

initgraph(&gd,&gm,"C:**\\**TC**\\**BGI");

color = getpixel(0, 0);

sprintf(array,"color of pixel at (0,0) = %d",color);

outtext(array);

getch();

closegraph();

return 0;

}