Lecture (12): Graphic Programming in C++ (part 2)

The "cleardevice" Function:

The "cleardevice" function is used to clear the screen in graphics mode. It is similar to "clrscr" function that is used to clear the screen text mode. Its syntax is:

```
cleardevice();
```

Closing Graphics Mode

The "closegraph" function is used to restore the screen to the text mode. When graphics mode is initialized, memory is allocated to the graphics system. When "closegraph" function is executed, it de-allocates all memory allocated to the graphics system. This function is usually used the end of the program. Its syntax is:

```
closegraph();
```

Text in Graphics Mode

In graphics node: text can also be written in different fonts, styles. sizes, colors, and directions. The graphic functions commonly used to create and print text are described below.

The "outtext" Function

The "outtext" function is used to print text on the computer screen in graphics mode. The text is printed at the current cursor position on the screen. Its syntax is:

```
outtext(string);
```

Where:

String: Represents the characters that are to be printed on the screen. It may be a string variable or string constant. The string constant is enclosed in double-quotes.

Example how to use cleardevice, closegraph and outtext function and print Programming digest into C++ graphics mode:

```
#include<graphics.h>
#include<conio.h>
main(){
int d, m;
d=DETECT;
initgraph (&d, &m, “            ”);
cleardevice();
outtext(“Programming digest”);
getch();
closegraph();
}
```

The “moveto” Function

The “moveto” function is used to move the current cursor position to a specified location on the screen where the output is to be printed. It is similar to “gotoxy” function used in text mode. Its syntax is:

```
moveto (x, y);
```

Where:

X: Represents the x-coordinate of the screen. It is the horizontal distance in pixels from the left of the screen. It may be an unsigned int type value or variable. For VGA, its value is from 0 to 639.

Y: represents the y-coordinate of the scren. It is the vertical distance in pixels from the top of the screen. It may be an unsigned int type value or variable. For VGA, its value is from 0 to 479.

Example how to use moveto function and print Programming digest into C++ graphics mode:

```
#include<graphics.h>
#include<conio.h>
main(){
int d,m;
d= DETECT;
initgraph(&d, &m, “    ”);
cleardevice();
moveto(400,200);
outtext(“Programming digest”);
getch();
closegraph(); }
```

The “outtextxy” Function

The “outtextxy” function is similar to the outtext” function but it is used to print text on the screen at a specified location. This function serves the purpose of both the “moveto” and “outtext” functions. Its syntax is:

outtextxy (x, y, string);

Where:

X: represents the x-coordinate of the screen. It is the horizontal distance in pixels from the left of the screen. It may be unsigned int type value or variable. For VGA, its value is from 0 to 639.

Y: represents the y-coordinate of the screen. It is the vertical distance in pixels from the top of the screen. It may be unsigned int type value or variable. For VGA, its value is from 0 to 479.

String: represents the string of characters that is to be printed on the computer screen. It may be a string variable or a string. constant. The string constant is enclosed in double quotes.

Example how to use outtextxy function and print Programming digest into C++ graphics mode:

```
#include<graphics.h>
#include<conio.h>
main(){
int d,m;
d= DETECT;
initgraph(&d, &m, “                    ”);
cleardevice();
outtextxy(100,200, “Programming digest”);
getch();
closegraph();
}
```

The “settextstyle” Function

The “settextstyle” function is used to define the text style in graphics mode. The text style includes the font type, font size and text direction. The syntax of this function is given as:

settextstyle (style, dir, size);

All the three parameters are of int type. These may be int type values or variables.

Where:

Style: specifies the font style. Its value range is from 0 to 10.

Dir: specifies the direction of the text in which it is to be displayed. Its value is from 0 to 1. It may be a numerical constant identifier. It is HORIZ DIR (for horizontal direction) or VERT_DIR (for vertical direction).

Size: specifies the font size of the text. Its value is from 0 to 72.

Example how to use settextstyle function and print Programming digest into C++ graphics mode:

```
#include<graphics.h>
#include<conio.h>
main(){
int d,m,c;
d= DETECT;
initgraph(&d, &m, “            ”);
cleardevice();
for(c=0; c<=10; c++) {
setextstyle(c,0,0);
outtextxy(100,20+c*20, “Programming digest”)   }
getch();
closegraph();
}
```

The “setcolor” Function

The setcolor” function is used to define color of the objects and the text in graphics mode. Its syntax is:

```
setcolor (co);
```

The “setbkcolor” Function

The “setbkcolor” function is used in graphics mode to define the background color of the screen. Its syntax is:

```
setbkcolor(co);
```

Where:

Co: Specifies the color. It may be it type value or variable. For VGA. Its value is from 0 to 15. It may also be numerical constant identifier eg. blue, green and red etc.

Example how to use setcolor and setbkcolor function and print Programming digest into C++ graphics mode:

```
#include<graphics.h>
#include<conio.h>
main(){
int d,m,co;
d= DETECT;
initgraph(&d, &m, “”);
cleardevice();
for(co=0; co<=15; co++){
setbkcolor(co);
setcolor(co+1);
settextstyle(0,0,2);
outtextxy(100,10+co*20, “Programming digest”);
outtextxy(200, 200,”press any key….”);
getch();  }
closegraph();  }
```

The “setlinestyle” Function

The “setlinestyle” function is used to set the line style of different objects that are drawn. The lines of these objects can be drawn in different styles, patterns and thickness. These attributes of the line are defined through this function. Its syntax is:

```
setlinestyle (style, pattern, thickness);
```

It has three parameters all of int types. These may be int type values or variables.

Where

Style: specifies the line style. Its value may be from 0 to 4.

Pattern: specifies the line pattern. Its value may be from 0 to 12.

Thickness: specifies the thickness of line. Its value may be from 0 to 3.

Example how to make stylish lines in c++ using setlinestyle function in graphic mode:

```
#include<graphics.h>
#include<conio.h>
main(){
int d,m,y=100,i;
d= DETECT;
initgraph(&d, &m, “                 ”);
cleardevice();
outtextxy(100,80,”lines with different styles”);
for (i=0, y=100; i < 4; y+=10,i++ ){
setlinestyle(i,0,0);
line (100,y,400,y);}
 outtextxy(100,150,”lines with different pattern”);
for(i=0, y=170; i < 12; y+=10,i++ ){
setlinestyle(i=0,i,0);
line(100,y,400,y); }
 outtextxy(100,300,”lines with different thickness”);
for(i=0, y=320; i < 3; y+=10,i++ ) {
setlinestyle(i=0,0,i);
line(100,y,400,y);}
getch();
closegraph();}
```