$$\therefore \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 35 \\ 6 \\ 15 \\ 1 \end{bmatrix} \qquad \therefore C'(35,6,15)$$

→ OpenGL geometric transformations functions —

* glTranslate — specifies translation parameters.
* glRotate — specifies parameters for rotation about any axis through the origin
* glScale — specifies scaling parameters w.r.t coordinate origin.
* glMatrixMode — specifies current matrix for geometric-viewing transformations, project transformations, texture transformations, or color transformations
* glLoadIdentity — sets current matrix to identity.
* glPushMatrix — copies the top matrix in the stack and store copy in the second stack position.
* glPopMatrix — Erases the top matrix in the stack and moves the second matrix to the top of the stack.
* glPixelzoom — specifies two-dimensional scaling parameters for raster operations.

```
void   circle_draw (GLint h, GLint k, GLint r)
{
    GLint d = 1-r, x = 0 , y = r;
    while (y>x)
    {
        plotpixels (h,k,x,y) ;
        if (d<0)  d+  = 2*x+3 ;
        else
```

——— × ———

→ Color Models :—

* Additive color —

ⓐ primary colors add together to get the perceived color. Eg — projectors and slide.

ⓑ with additive color, primaries add light to an initially black display, yielding the desired color.

ⓒ primary colors — R, G, B.

* Subtractive color —

ⓐ color pigments remove color components from light that is striking the surface.

ⓑ colors — C, M, Y (cyan, magenta, yellow).

* characteristics of colors —

ⓐ Hue — dominant frequency

ⓑ Brightness — intensity of the light

ⓒ Saturation — purity of the color.

— combination of RGB = white

— combination of CMY = black

* conversion from RGB to CMY :

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = 1 - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

* conversion from CMY to RGB :

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = 1 - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}.$$

→ Illumination Model :—

* Light source parameters —

ⓐ positions
ⓑ electromagnetic spectrum
ⓒ shape

* Surface Parameters —

ⓐ Position
ⓑ Reflectance properties
ⓒ Position

* Eye (camera) parameters —

ⓐ position ⓑ sensor spectrum activities.

→ Phong Illumination Model :—

* Ambient —

$\boxed{I = I_a * k_a.}$, where

I = intensity,
I_a = intensity of ambient light
k_a = Ambient Reflection co-efficient
$0 \leq k_a \leq 1$

Date
Page

* Diffuse —

$I = I_s * K_d \cos\theta.$

$I = I_s * K_d\, N.S.$

$\cos\theta = N.S.$

$$\left[\begin{array}{c} 0 \leq K_a \leq 1 \\ 0 \leq \theta \leq 90 \end{array}\right].$$

* Specular —

$I = I_s * K_d \cos^n \alpha.$

n — specular intensity.

$\cos\alpha = R.V.$

I = Ambient + Diffuse + Specular

MODULE — 4

→ OpenGL 3D Viewing Functions :—

* gluLookAt — specifies 3D viewing parameters.

* glOrtho — specifies parameters for a clipping window and the near & far clipping plane for an orthogonal projection.

* gluPerspective — specifies ~~view of~~ field-of-view angle and other parameters for a symmetric perspective projection

* glFrustum — specifies parameters for a clipping window and near & far clipping planes for a perspective projection.

* glClipPlane — specifies parameters for an optional clipping plane.

→ Visibility surface Detection Algorithm :—

* Object-space methods — (Back-face detection algorithm) compares parts of objects to each other to determine which surfaces should be labelled as visible.

* Image space methods — (depth buffer algorithm) visibility is decided point by point at each pixel position on the projection plane.

→ Open GL Visibility detection functions :—

* glCullFace — specifies front or back planes of polygons for culling operations when activated with glEnable (GL CUEL FACE).

* glutInitDisplayMode — specifies depth-buffer operations using argument.

* glClear (GL DEPTH BUFFER BIT) — initialises depth-buffer values to the default (1.0) or a value specified by the glClearDepth function

* glClearDepth — specifies an initial depth-buffer value.

Q. Write and explain the process in interaction of input device.

→ • Input devices contain a trigger which can be used to send a signal to the operating system
  - button on mouse
  - pressing or release a key

• When triggered, input devices return information (their measure) to the system
  - mouse returns position information.
  - keyboard return ASCII code.

• Request Mode
  - input provided to program only when user triggers the device.
  - typical of keyboard input.

• Event Mode
  - most systems have more than one input device, each of which can be triggered at an arbitary time by a user.
  - each trigger generates an event whose measure is put in an event queue which can be examined by the user program