UNIT : 3

Windowing & Clipping

The graphics packages allow a user to specify what portion of the image is to be displayed and where the images is to be displayed.

A convenient co-ordinate system which is used to define a reference frame of the picture forms the world co-ordinates.

* A world-co-ordinate area selected for display is called a window. (OR) A rectangular area specified in world co-ordinates is called a window.

* An area on a display device to which a window is mapped is called a viewport.

The window defines what is to be viewed & the viewport defines where it is to be display.

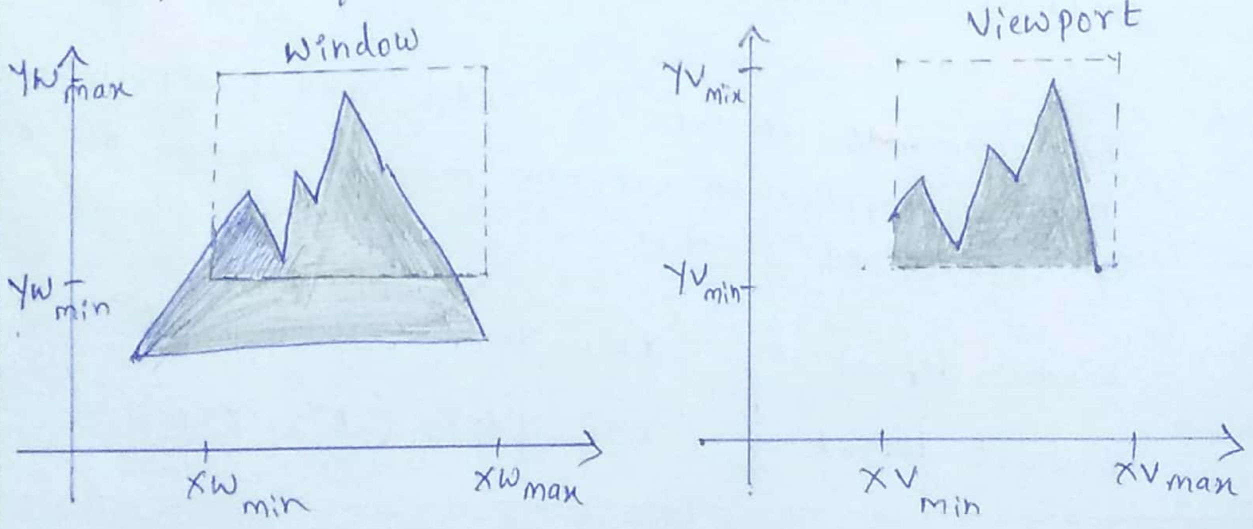
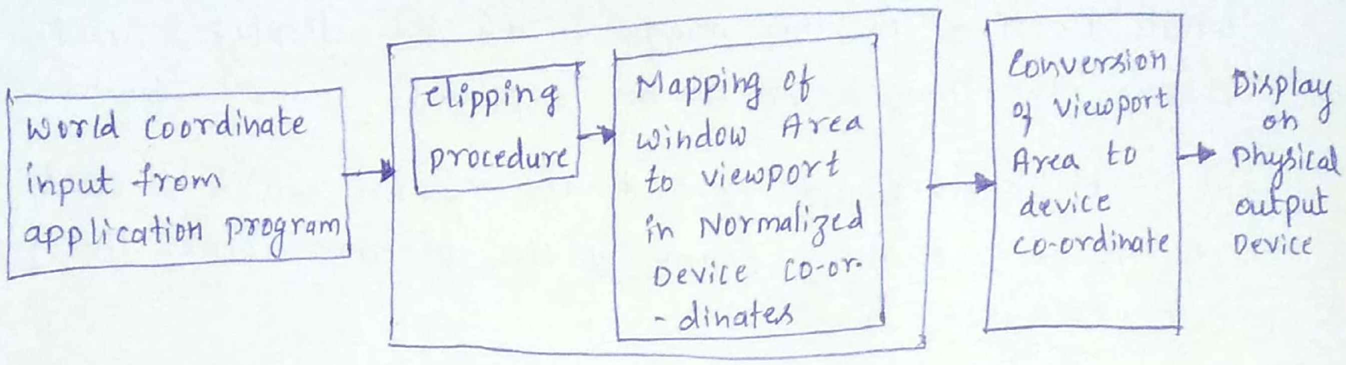


Fig : A viewing transformation using rectangles for the window and viewport.

A mapping of a part of a world-co-ordinate scene to device coordinates is called viewing-transformation or window-to-viewport transformation.

* 2-D Viewing transformation



World coordinate input from application program
clipping procedure
Mapping of window Area to viewport in Normalized Device co-or-dinates
Conversion of Viewport Area to device co-ordinate
Display on Physical output Device

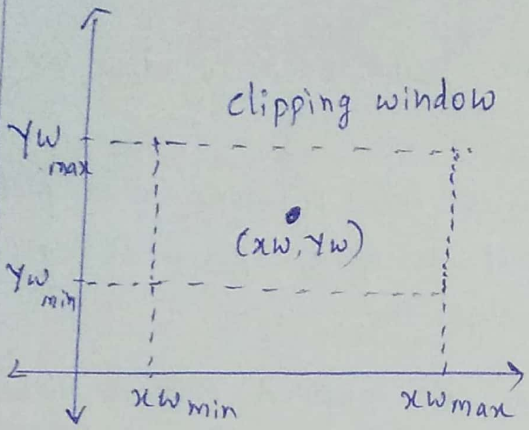
- To obtain a particular orientation for the window, we can set up a two-dimensional viewing co-ordinates system in the world co-ordinate plane.
- All parts of the picture that lie outside the viewport are are clipped and the contents of the viewport are transferred to device co-ordinates.
- clippings are used in drawing packages to eliminate parts of a picture inside or outside of a designated screen area.
- The commands to set the window and view port areas from an application program may be defined as set of pre-defined function.

window defined by $(xw_1, yw_1), (xw_2, yw_2)$

Viewport defined by $(xv_1, yv_1), (xv_2, yv_2)$

* Window - to - Viewport transformation.

The main objective of the window to viewport mapping is to convert the world co-ordinate (xw, yw) of any point to its normalized device coordinates (xv, yv).



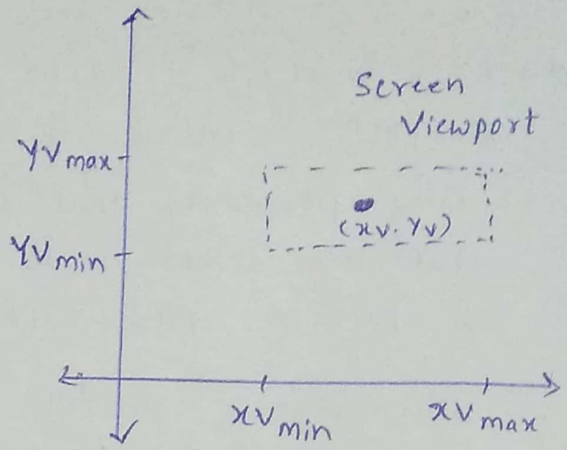


Fig. A point $(xw\ yw)$ in window is mapped to viewport at (xv, yv)

Both window & viewport is described by 4 co-ordinates

A window is specified xw_{min}, xw_{max}, yw_{min} yw_{max}.

A viewport is specified xv_{min} xv_{max}, yv_{min} yv_{max}.

$$\boxed{\begin{aligned} xv &= xv_{min} + (xw - xw_{min}) * sx \\ yv &= yv_{min} + (yw - yw_{min}) * sy \end{aligned}}$$

where

$$sx = \frac{(xv_{max} - xv_{min})}{(xw_{max} - xw_{min})}$$

$$sy = \frac{yv_{max} - \cancel{x}v_{min}}{yw_{max} - yw_{min}}$$

The conversion is performed with the following transformation

1. Perform a scaling transformation using fixed point position of $(xw_{min} \cdot yw_{min})$ that scales the window area to the size of the viewport.
2. Translate the scaled window area to the position of the viewport.

* Clipping Operations

***→ Elimination of parts scene outside a window or viewport is called clipping.

→ Any procedure that identifies those portions of a picture that are either inside or outside of a specified region of space is called clipping algorithm.

→ whenever a window area is mapped onto a viewport, this results in the display of only those parts of the picture within the window. All other portions of the picture which are outside the window are discarded.

* Applications of clipping :

→ Extracting parts of a defined scene for viewing.
→ Locating visible surfaces in 3D views.
→ Anti- aliasing line segments or boundaries of objects.
→ Creating objects using solid modeling procedures.
→ Displaying a multiwindow environment.
→ Drawing and painting the parts of a picture object.

*** * Different types of clipping

1. Point clipping.
2. Line clipping.
3. Polygon Clipping.
4. Curve clipping.
5. Text clipping.

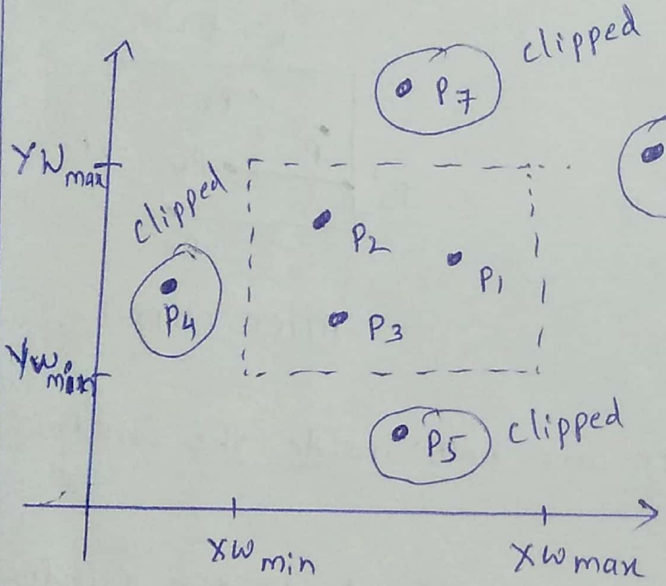
1. Point Clipping :

clipping of points is done by comparing each point with the boundary of the rectangle window.

Point clipping against a window specification is done by testing the co-ordinate values to determine whether they are within the boundaries or not. using the following conditions.

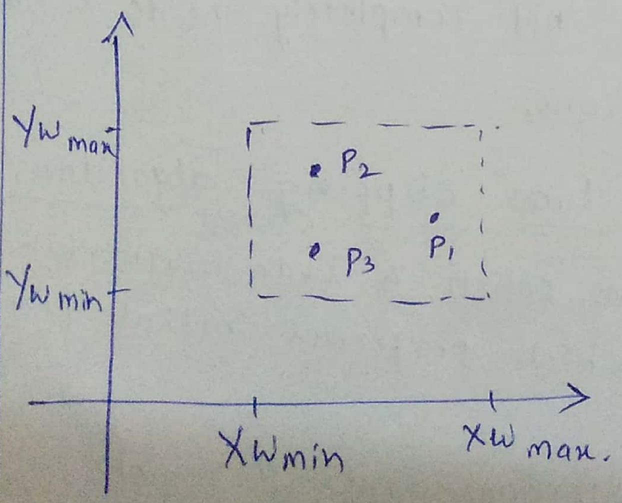
$$\boxed{\begin{array}{l} xw_{min} <= x <= xw_{max} \\ yw_{min} <= y <= yw_{max} \end{array}}$$

The point (x, y) lies inside the clip area.



points P4, P5, P6, P7 are outside the window are clipped.

After point clipping



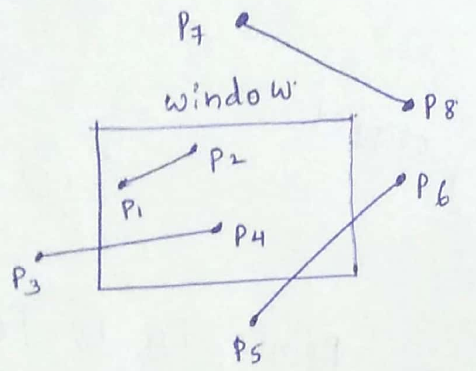
points P1, P2. P3 are within the window are not clipped.

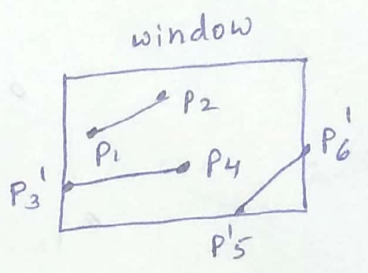
2. Line clipping.

A line clipping procedure

- Test a given line segment whether it lies completely inside the clipping window
- Determine whether it lies completely outside the window
- If we can not identify a line completely inside or outside, then perform intersection calculations.

Exa :-





After clipping.

Before clipping.

Line P_1 & P_2 end points are both inside the window, so it not clipped.

Line P_7 & P_8 end points are both outside the window, so it is completely clipped.

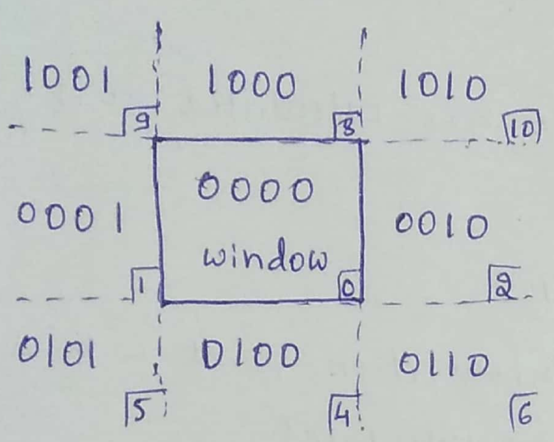
Line P_3 & P_4, P_5 & P_6 are not completely inside & outside ∴ it requires calculations.

★ Cohen - Sutherland Line clipping algorithm :

It is invented by Dan cohen & Ivan sutherland. It is the efficient algorithm which performs initial tests on a line.

The algorithm divides a 2D space into 9 regions, of which only the middle part (viewport) is visible.

→ Every line end point in a picture is assigned a 4-digit binary code, called a region code



4 bit	3 bit	2 bit	1 bit
above	below	right	left

All lines fall into one of three categories.

1. Both endpoints lie inside the window → Accept.
2. Both endpoints outside the window → reject
3. Neither 1 nor 2 then clip part of line outside the borders.

Algorithm :-

1. A 4-bit region code number is assigned to an endpoint (X, Y).
2. Once the region codes for all line endpoints are known, then we can determine which line are completely inside the window or outside the window.
3. The lines within the window have a region code of 0000 for both endpoints. accept the line.
4. The lines which outside the window are rejected.
5. Perform the Logical AND operation with both region codes.

7. If the result is not 0000, it means line is completely outside the window.. Rejected.

8. If the result is 0000, the line is neither inside nor outside

9. Then Intersection points are calculated using equation parameters.

For a line (x_1, y_1) & $(x_2\ y_2)$.

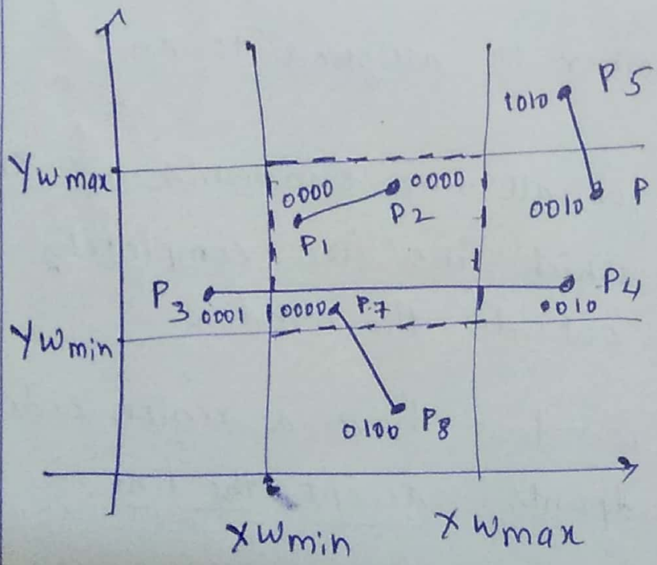
The x co-ordinate of the intersection point with a horizontal boundary can be calculated

$$\boxed{x = x_1 + \frac{(y - y_1)}{m}}$$ wher $y = yw_{min}$ or yw_{max}.

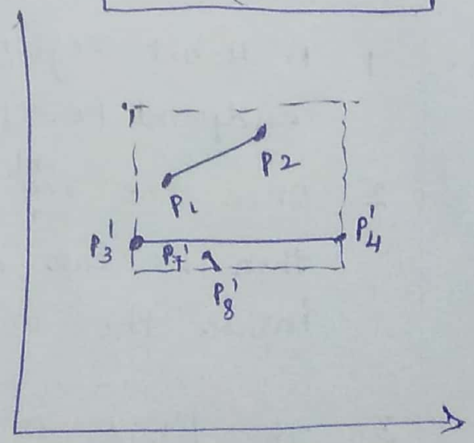
The y co-ordinate of the intersection point with a ~~horizont~~ vertical boundary can be calculated

$$\boxed{y = y_1 + m\,(x - x_1)}$$ wher $x = xw_{min}$ or xw_{max}.

slope of the line $\boxed{m = \frac{(y_2 - y_1)}{(x_2 - x_1)}}$



Before clipping



After clipping

$\boxed{P_1 \& P_2}$

Logical and

$$\begin{array}{r} 0000 \\ 0000 \\ \hline 0000 \end{array}$$

Not clip

$\boxed{P_5 \& P_6}$

$$\begin{array}{r} 1010 \\ 0010 \\ \hline 1000 \end{array}$$

Clip.

$\boxed{P_3 \& P_4}$

$$\begin{array}{r} 0001 \\ 0010 \\ \hline 0000 \end{array}$$

Not clip

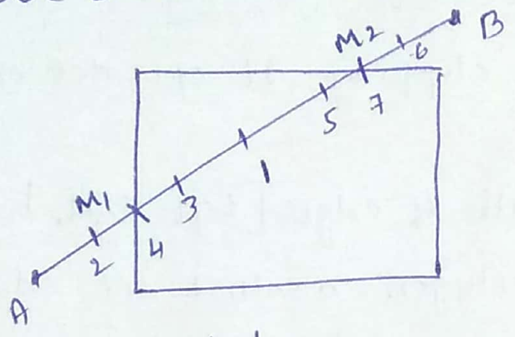
$\boxed{P_7 \& P_8}$

$$\begin{array}{r} 0000 \\ 0100 \\ \hline 0000 \end{array}$$

Not clip.

Scanned by CamScanner

* Mid point subdivision algorithm :



window.

The line is divided at its midpoint into 2 shorter line segments. The clipping categories of the 2 new line segment are next computed using the region codes.

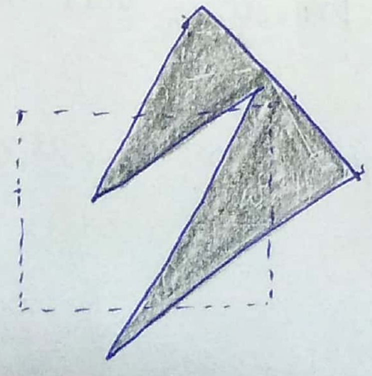
The line can be

1. completely in the visible region.
2. Not visible and lie outside the window
3. Line is again divided into shorter segment & categorized.

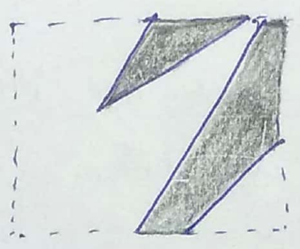
3. Area clipping or Polygon clipping :

To clip polygons, we cannot use the line alipping algorithm, need to modify the line clipping procedures.

The output of a polygon clipper should be a sequence of vertices that defines the clipped polygon boundaries.



Before clipping

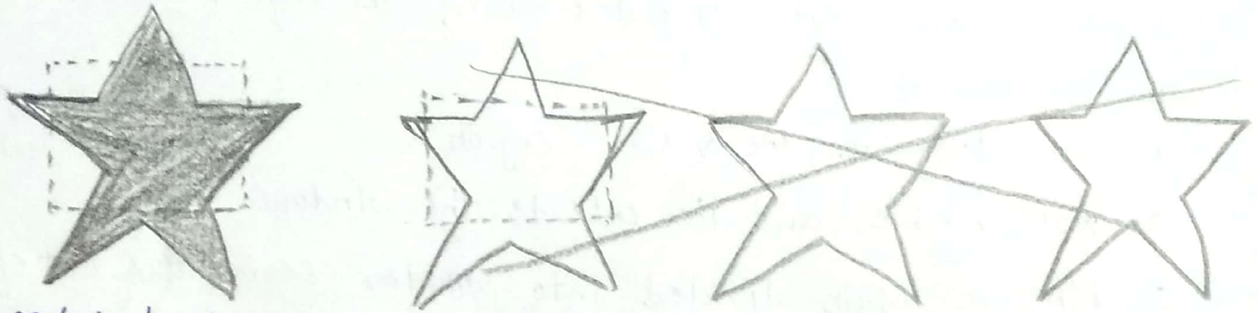


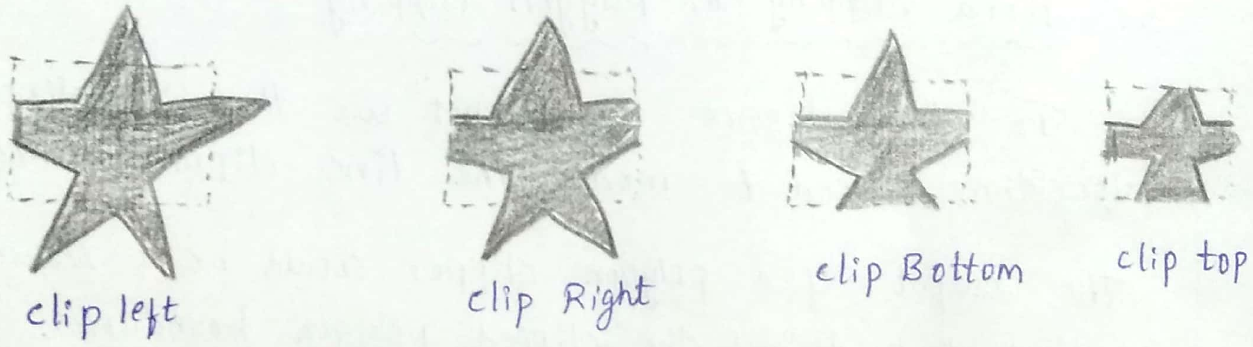
After clipping.

* Sutherland - Hodgeman polygon clipping :

It is user for polygon clipping. It operates on the vertices of the polygon.

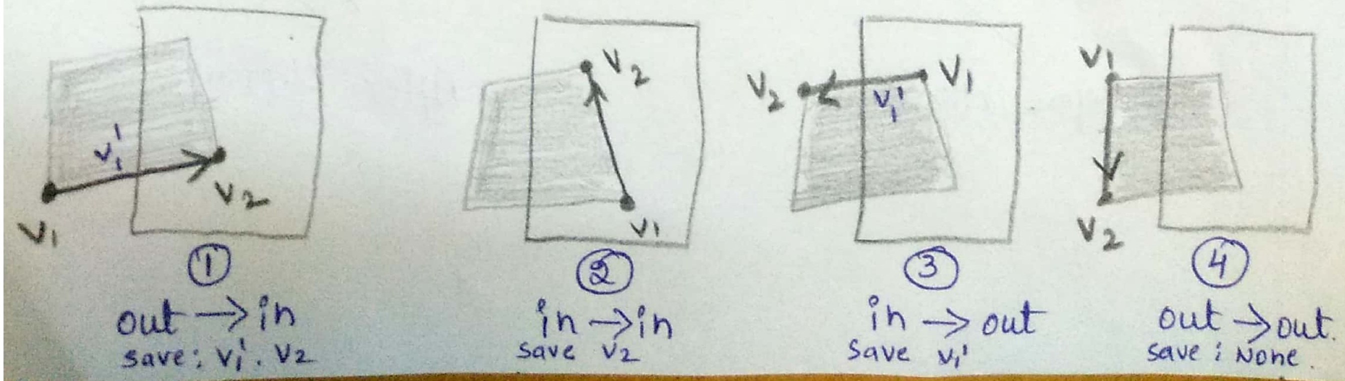
It clip the polygon on all 4 edges [Left, Right, bottom, top] by clipping the entire polygon against one edge, then taking the resulting polygon and clipping against 2nd edge and so on for all 4 edges.





There are 4 possible cases when processing vertices in sequence.

The 4 cases are illustrated in figure for successive pair of polygon vertices,



case 1 : If first vertex is outside the window, second vertex is inside, both intersect point of polygon edge & second vertex is saved.

Save : V_1' , V_2.

case 2 : If both vertices are inside the window boundary, only second vertex is added (saved) Save : V_2

case 3 : If the first vertex is inside, second vertex is outside only the edge intersection with the window boundary is added (saved). Save : V_1'

case 4 : If both input vertices are outside the window boundary, nothing is saved. Save : None.

4. Text clipping :

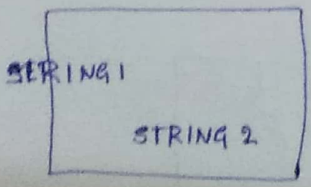
There are several techniques that can be used to provide text clipping.

a) String clipping.

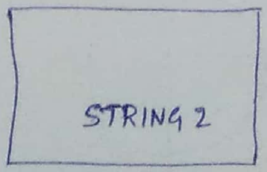
b) Character clipping.

c) Individual component clipping.

a] String clipping : If all of the string is inside a window it accepted, otherwise the string is discarded.

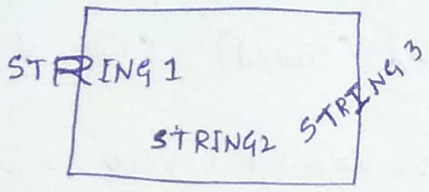


Before clipping

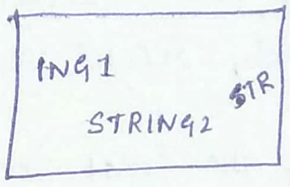


After clipping.

b) Character clipping : It discard only those characters that are not completely inside the window. Any character that either overlaps or is outside a window boundary is clipped.

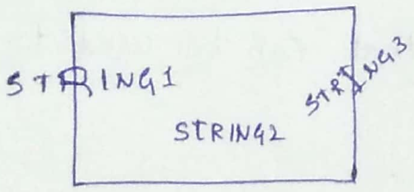


Before clipping

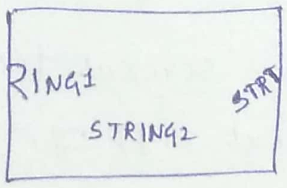


After clipping.

c) Individual component clipping : If an individual character overlaps a clip window boundary, it clip off the parts of the character that are outside the window.



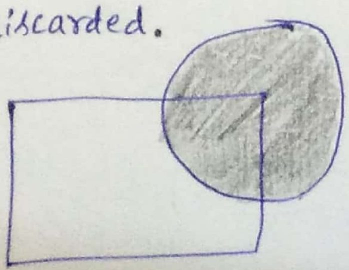
Before clipping

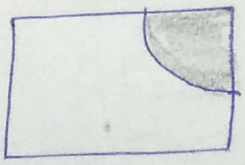


After clipping.

5. Curve clipping :

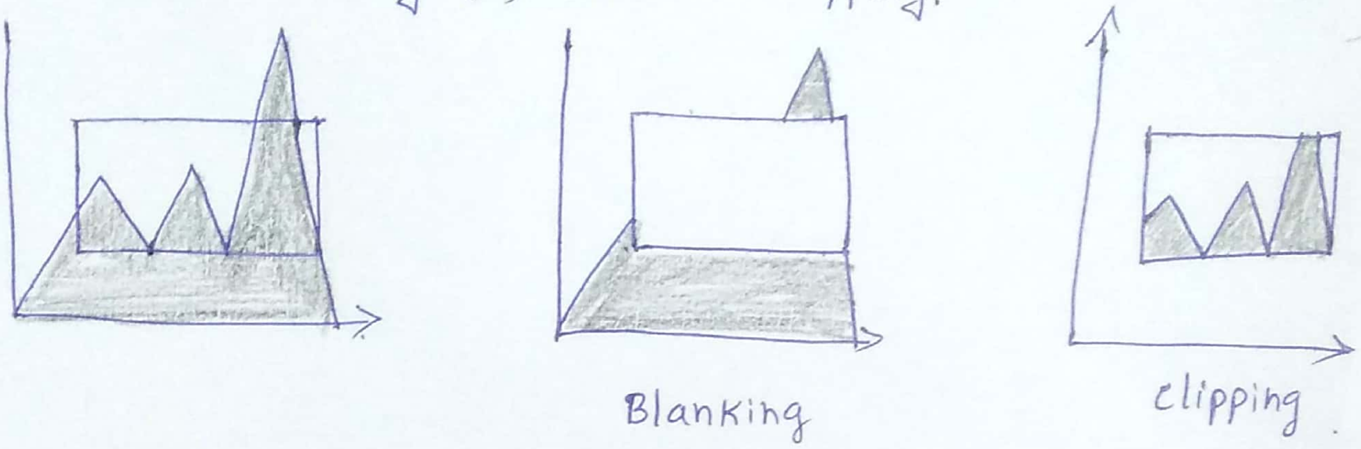
The bounding rectangle for the curved object is first tested for overlap or intersection with the rectangular window, If the bounding rectangle for the curve is completely within the window then the object is saved otherwise discarded.


Before clipping


After clipping.

Blanking (or) Exterior clipping

All the parts outside the region must be retained but within the window boundary region is discarded is called blanking (or) Exterior clipping.



what do you Mean by device-cordinate ?

device co-ordinates used to place the object on the display device.