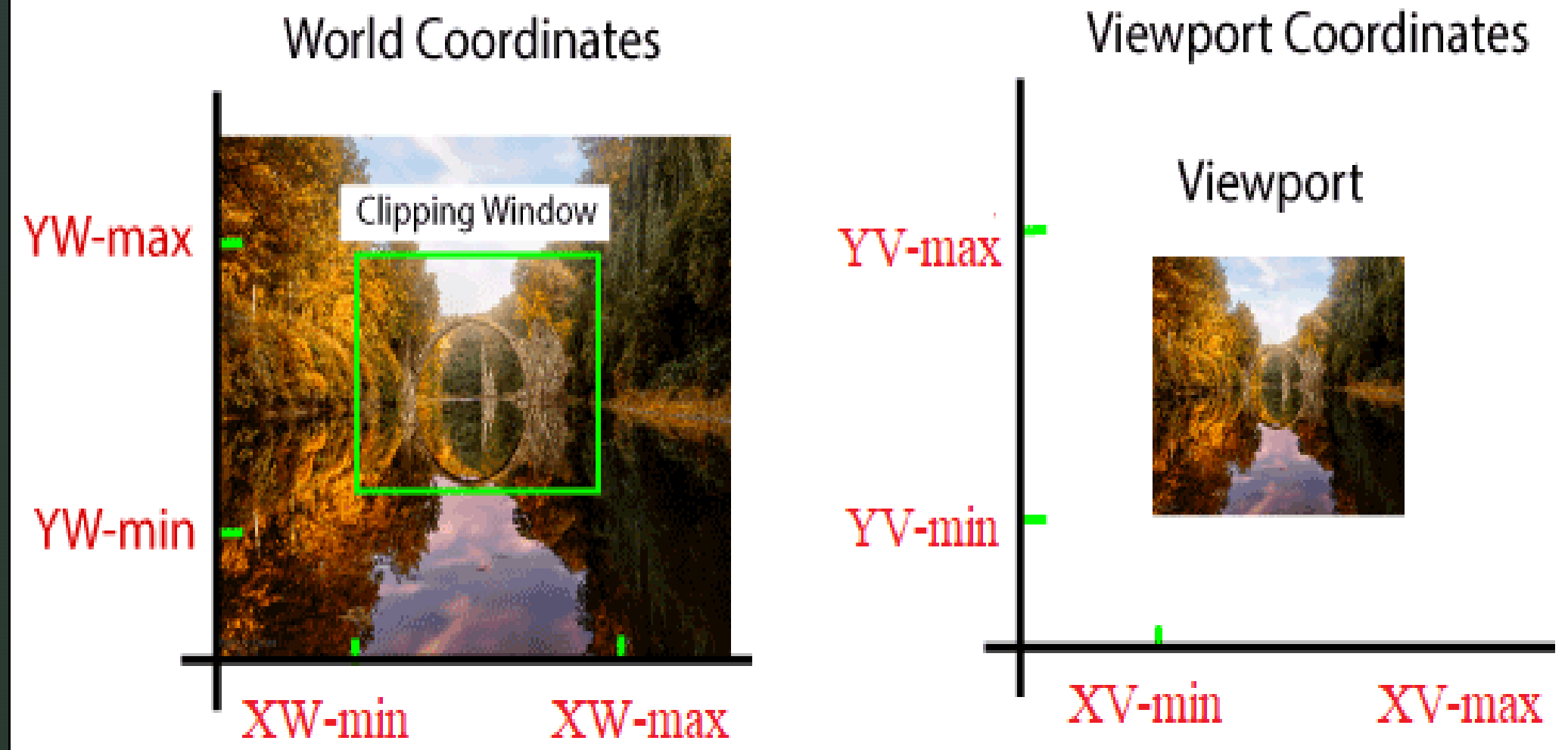
Computer Graphics Window

- “The Capability to show some part of an object in a window is known as windowing.”
- “The rectangular area describes in the world coordinate system is called the window.”
- **Viewport:**
- “The viewport can be defined as an area on the screen which is used to display the object.” The window is an area space for the object. Viewport surrounds the object. We need the coordinate transformation to display the object on the screen. We can define many viewports on a different area of the screen and also see the same object from a different angle in the viewport.

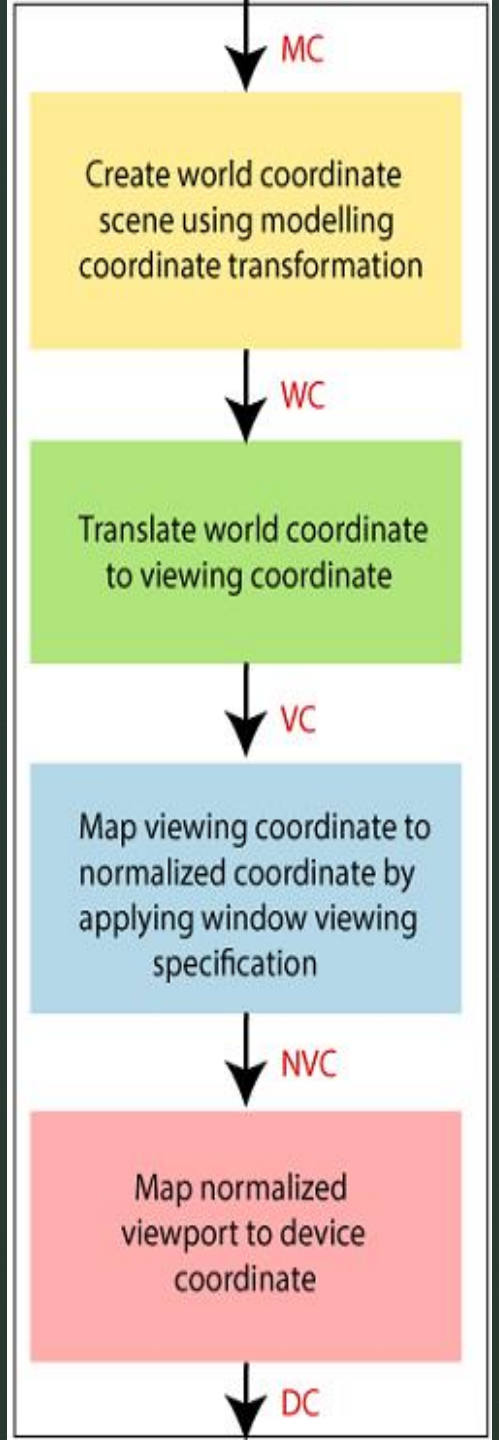

World Coordinates
Clipping Window
YW-max
YW-min
XW-min
XW-max
Viewport Coordinates
Viewport
YV-max
YV-min
XV-min
XV-max

Window to Viewport transformation:

- Window to viewport transformation is a process of converting two-dimensional or world into a device coordinate.”
- The object inside the clipping window is mapped to the viewport. The viewport is displayed inside the interface window on the screen. We can use the clipping window to select the part of an object, and the viewport is used to display the selected part of the object on the screen or output device.

Steps for Window to Viewport Transformation: We can follow the following steps for transform window to viewport:

- **Step 1: Translation of the window towards the Origin**– If we shift the window towards the origin, the upper left, and the lower-left corner of the window will be negative (-). The translation factor also should be negative (-).
- **Step 2: Resize the window to viewport size**– To Convert the size of the window into the size of the viewport, we will use the following formulas:
- $S_x = XV_{max} - XV_{min} / XW_{max} - XW_{min}$
- $S_y = YV_{max} - YV_{min} / YW_{max} - YW_{min}$
- **Step 3: Translation of window (Position of window and viewport must be same)**– If the lower-left corner of viewport is (0, 0), then the window lower-left corner is already shifted on origin after following step 1.
- If the lower-left corner is not equal to (0, 0), then the translation factor should be positive (+).


MC
Create world coordinate scene using modelling coordinate transformation
WC
Translate world coordinate to viewing coordinate
VC
Map viewing coordinate to normalized coordinate by applying window viewing specification
NVC
Map normalized viewport to device coordinate
DC

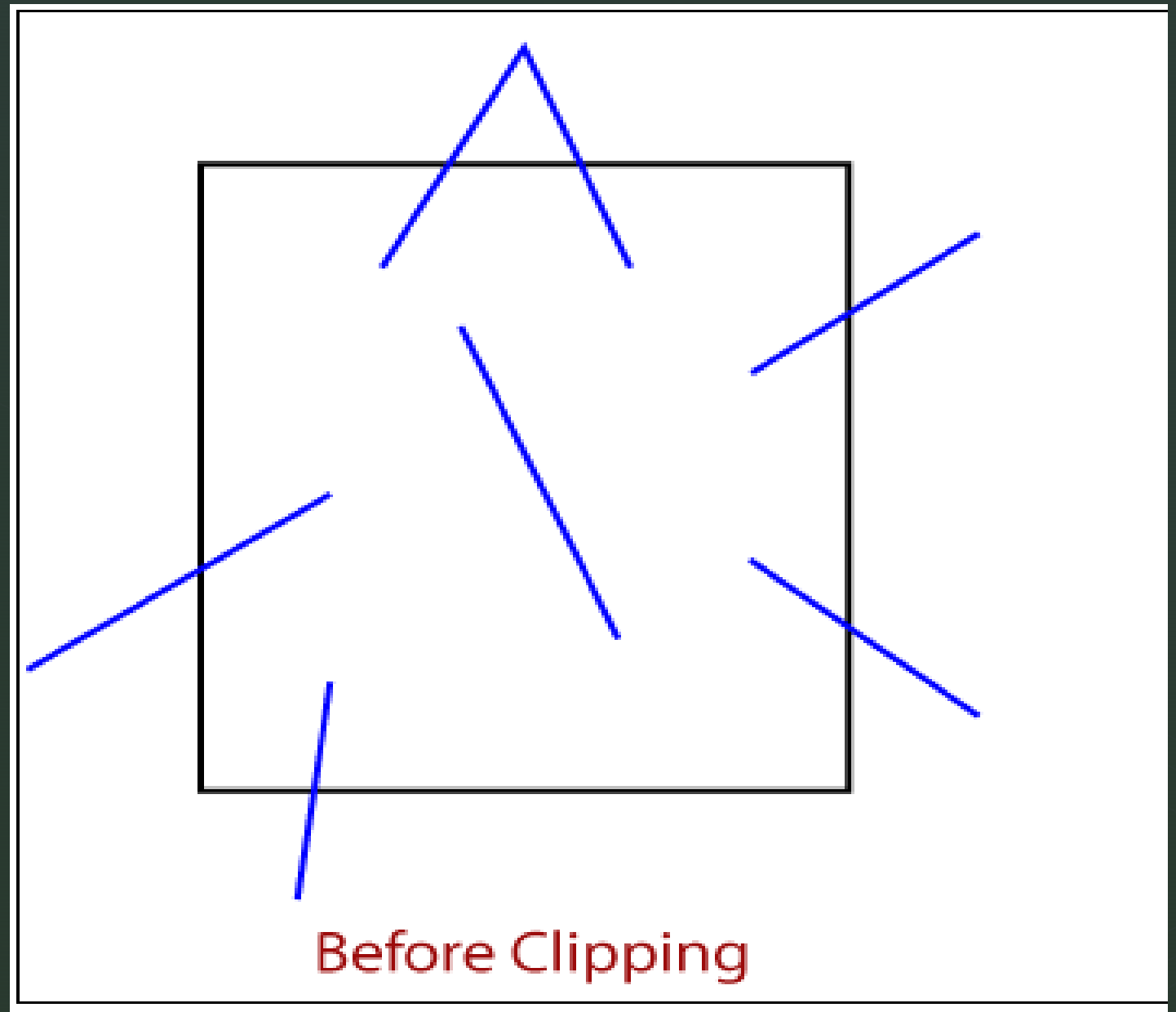
- It may be possible that sometimes the size of viewport is greater or smaller than the window. **In that situation, we have to enlarge or compress the size of the window according to the viewport. We can perform it using some mathematical calculations.**
- A point on window = (XW, YW) Corresponding point on viewport = (XV, YV)
- **To calculate (XV, YV) –**
- Normalized point on window = $(XW- XW_{min} / XW_{max} – XW_{min})$ $(YW- YW_{min} / YW_{max} – YW_{min})$
- Normalized point on viewport = $(XV- XV_{min} / XV_{max} – XV_{min})(YV- YV_{min} / YV_{max} – YV_{min})$
- **Now, the position of object in viewport and window are same-**
- **For Coordinate x:**
- $(XW- XW_{min} / XW_{max} – XW_{min}) = (XV- XV_{min} / XV_{max} – XV_{min})$
- **For Coordinate y:**
- $(YW- YW_{min} / YW_{max} – YW_{min}) = (YV- YV_{min} / YV_{max} – YV_{min})$
- Now, we get
- $XV = XV_{min} + (XW- XW_{min}) S_x$ and $YV = YV_{min} + (YW- YW_{min}) S_y$
- Here S_x and S_y are the scaling factor for x and y coordinate.
- $\mathbf{S_{x} = XV_{max} – XV_{min} / XW_{max} – Xw_{min}}$ **and** $\mathbf{S_{y} = YV_{max} – YV_{min} / YW_{max} – YW_{min}}$

Clipping in Computer Graphics

- “The Clipping is a type of transformation used in computer graphics to remove lines, objects, and segments of lines that are outside the computer screen or viewing pane.” The clipping is a process of deciding the visible and invisible part of the image, object, or any line segment.
- We only select the visible part and remove the invisible part of the line. The process clipping can be used in two-dimensional and three-dimensional graphics.

Applications of clipping

- We can use clipping for drawing operations.
- It is used for separating the important part of an image.
- The Solid modeling technique in clipping helps to build three-dimensional objects.
- We can perform various operations that correlate with the pointing of an object. **For example,** delete, copy, insert, and moving selected parts of an object, etc.
- Clipping helps us to describe the visible and invisible parts of 3D objects.
- **Types of lines**
- The lines are divided into three types.
- **Visible Line:** The line lying inside the view pane, is a visible line.
- **Invisible Line:** The line lying outside the view pane, is an invisible line.
- **Clipped Line:** **“**A line that lies inside or outside the window is called clipped line.**”** A point where the line cut the view pane is known as the Intersection point of the line


Before Clipping

Types of clipping

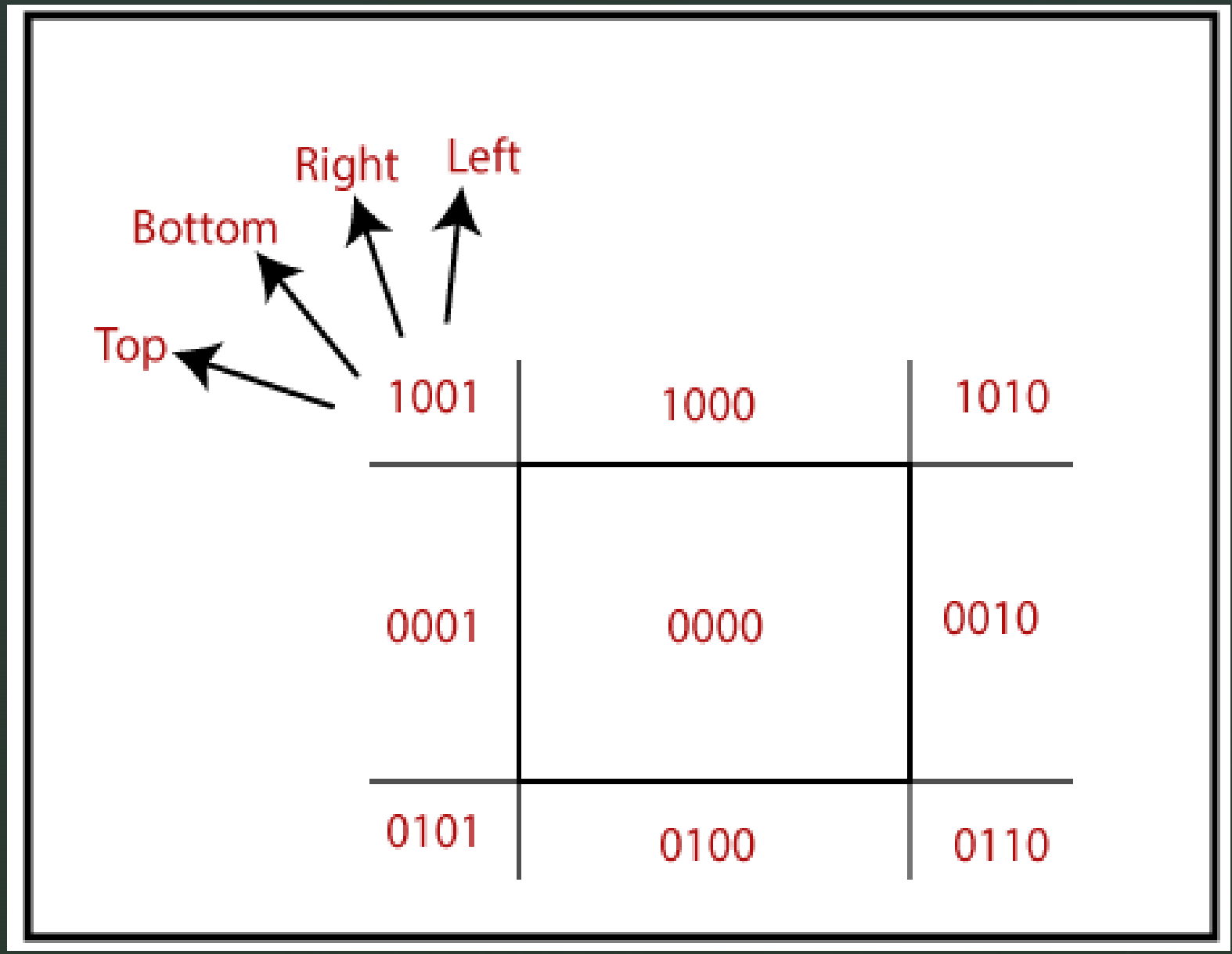
- We can divide clipping into following types-
  Line clipping
  Point clipping
  Text clipping
  Exterior Clipping
  Curve clipping
  Polygon clipping

Line Clipping in Computer Graphics

-
- The line clipping is a process in which we can cut the part of the line, which lies outside the view pane. Only those lines are visible, which lie inside the view pane.
- The line clipping process is implemented by following line clipping algorithms–
- Cohen Sutherland Line Clipping Algorithm
- Midpoint Subdivision Line Clipping Algorithm

Cohen Sutherland Line Clipping Algorithm

- In this algorithm, we will divide the view pane into nineequal segments (as shown in the below figure) that only serve the viewport.
- Now, we will represent the top, bottom, left, and right corner of the window with 4 bits. This 4bit can be described with the following point that:
- If an object lies within any particular corner position, that corner value will be 1, else it will be 0.
- The allocation of bits depends on **“TBRL”** (Top, Bottom, Right, Left) rule.
- Suppose, if the point of a line appears in the top-left corner, then according to TBRL, the value is 1001. We will allot the bits as-
- In this way, we check TBRL for each segment and allot the bits accordingly.


Right
Left
Bottom
Top
1001
1000
1010
0001
0000
0010
0101
0100
0110

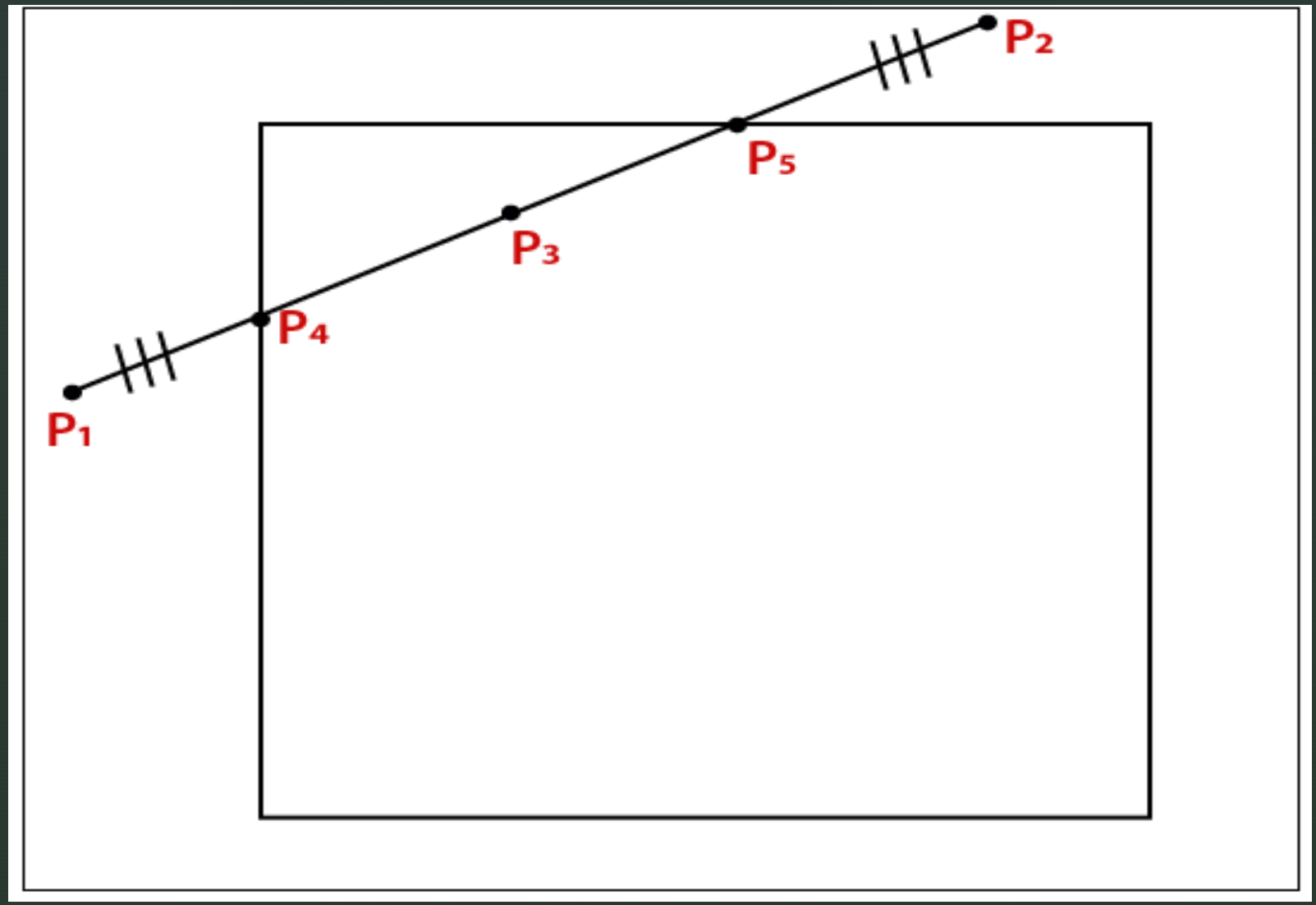
- **Visible Line:** When both points (starting and ending) of the line are entirely situated inside the window.
- **Invisible Line:** When both points (Starting and ending) of the line are completely situated outside the window.
- If we have (x_{min}, x_{max}) and (y_{min}, y_{max}) coordinates of the view pane (window).
- Then, it should be described as-
- $x_0, x_1 > x_{max}$
- $y_0, y_1 > y_{max}$
- $x_0, x_1 < x_{min}$
- $y_0, y_1 < y_{min}$
- **Clipped Line:** Everyline has two endpoints. Let (x_0, y_0) and (x_1, y_1) are points of the line. If one point of the line situated inside the window and the other one is outside the window, then the line is known as Clipped Line.

- **Advantages:**
- It is easy to use and implement.
- We can perform clipping and testing in a particular manner.
- It is a fast algorithm.
- **Disadvantages:**
- Sometimes it performs needless clipping.

Midpoint Subdivision Line Clipping Algorithm:

- The midpoint subdivision algorithm is used to clip the line. The algorithm is based on finding the midpoint of the line. We can divide the line into two equal parts. There should be following categories of the line-
- **Visible line**
- **Invisible line**
- **Partially visible**

We can calculate the midpoint of the line by the following formula- $p_m = (p_1 + p_2)/2$



- Use coen suterlend to determine visible, invisible and the partially visible lines. For partially visible line,
- **Step 1:** For partially visible line, we need to find the midpoint.
- $X_m = (x_1 + x_2)/2$ **(For x coordinate)**
- $Y_m = (y_1 + y_2)/2$ **(For y coordinate)**
- **Step 2:** Weneed to check that the line is near to the boundary of the window or not.
- **Step3** If the line is visible or invisible, then repeat steps 1 to 5.
- **Step 4:** Stop.

Point Clipping

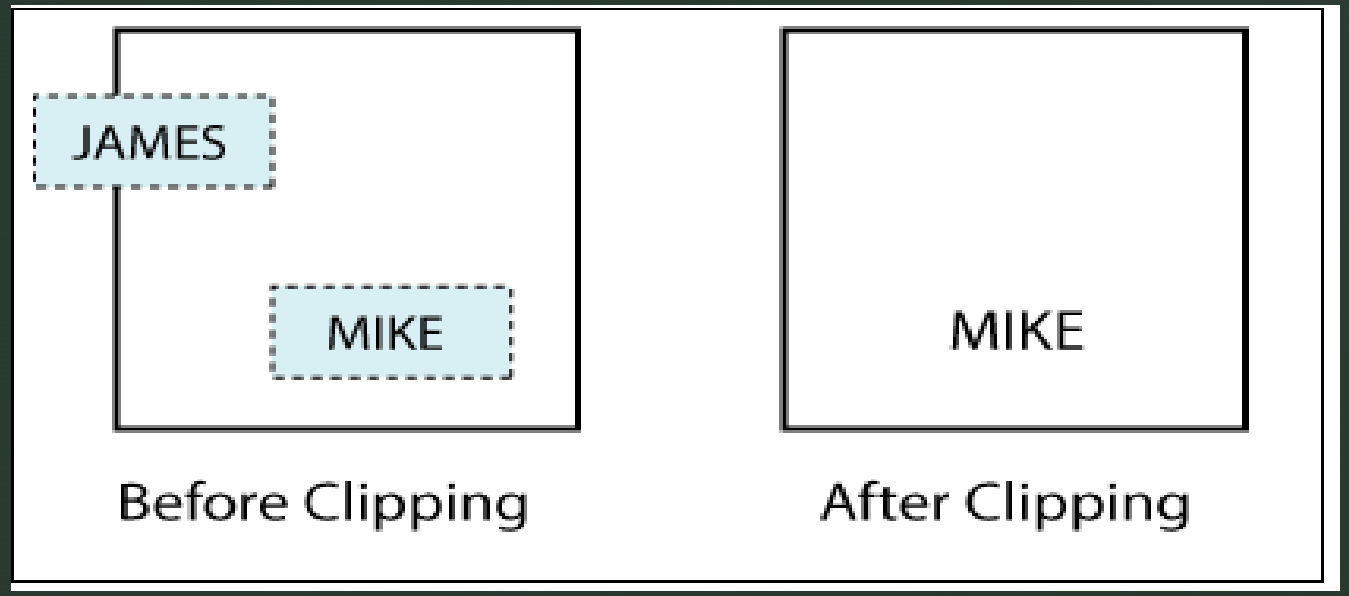
- "Point clipping is a process which is used to define the point position." The point is either inside the view pane (window) or outside the view pane.
- It includes two terms-
- **Window:** It means what to display?
- **Clipping:** It means discarding the portion that is outside the window.
- **Example:** Let us have a view pane (window). The coordinates of the window are-
- **(xw_{max}, xw_{min}) –** For X-axis of the window
- **(yw_{max}, yw_{min}) –** For Y-axis of the window
- Let us assume a point coordinate (P, Q). If the point lies inside the window, then there is no need to perform point clipping. But if the point lies outside the window, we need to perform clipping. We can understand it by the following equation-
- **xw_{max} <= P <= xw_{min}, yw_{max} <= Q <= yw_{min}**
- There are four conditions; if these four are satisfied, then the point lies inside the window. If anyone condition is not satisfied, then the point lies outside the window, it means we have to perform clipping on the point.
- **xw_{min} <= P, xw_{max} => P**
- **yw_{min} <= Q, yw_{max} => Q**

Text Clipping

- **“Text Clipping is a process in which we remove those part (portion) of string that is outside the view pane (window).”** Various methods and techniques can do the text clipping. These techniques depend on the character generation method. It means we can select text clipping techniques according to the text generation technique.
- **Method of Text Clipping**
  - **All or none string clipping**
  - **All or none character clipping**
  - **Text Clipping**

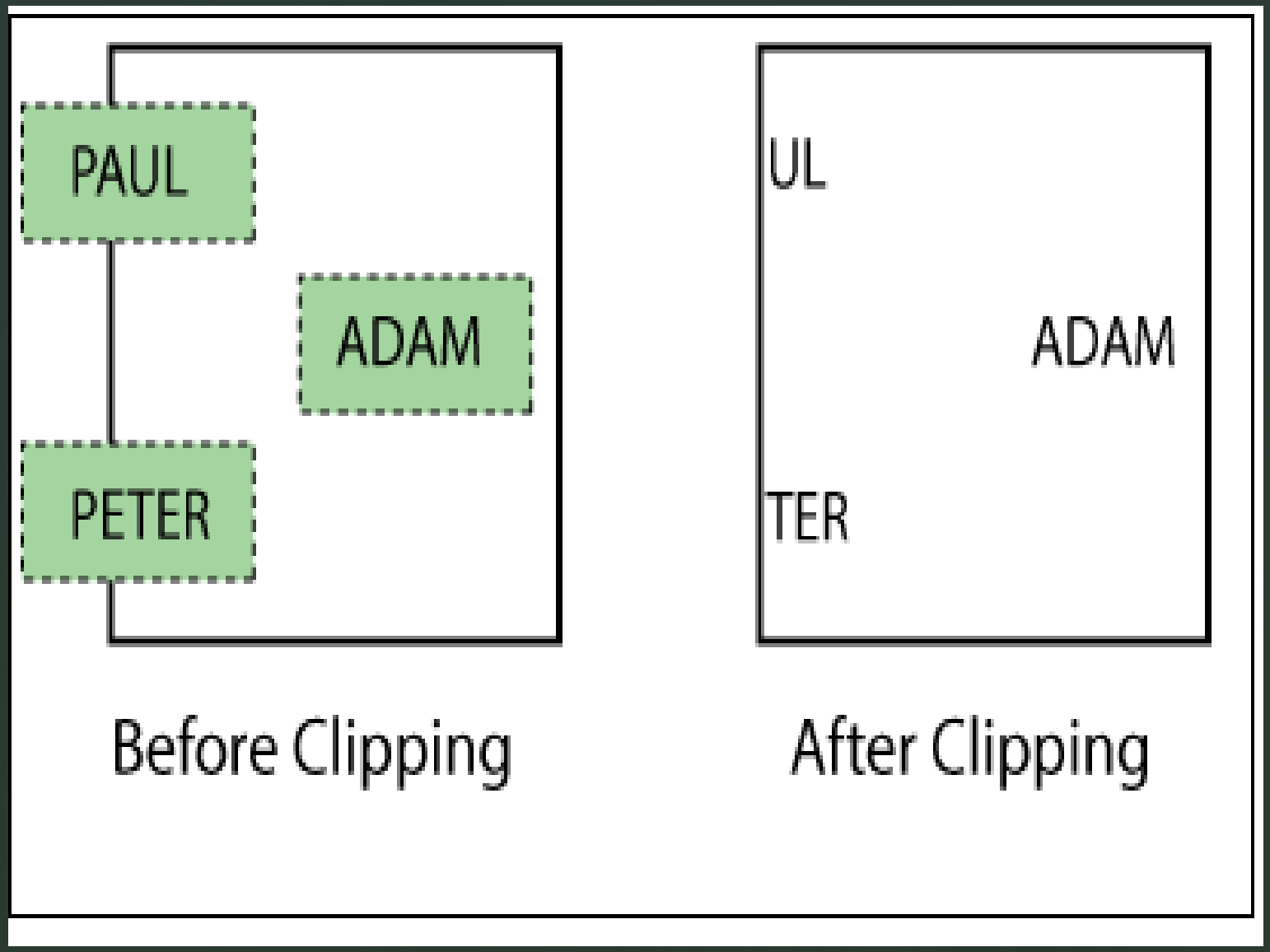
All or none string clipping

- **:** In this method, we only consider the string that is entirely inside the view pane (window). We remove the string that is partially or fully outside the boundary. We compare the window coordinates with string coordinates.



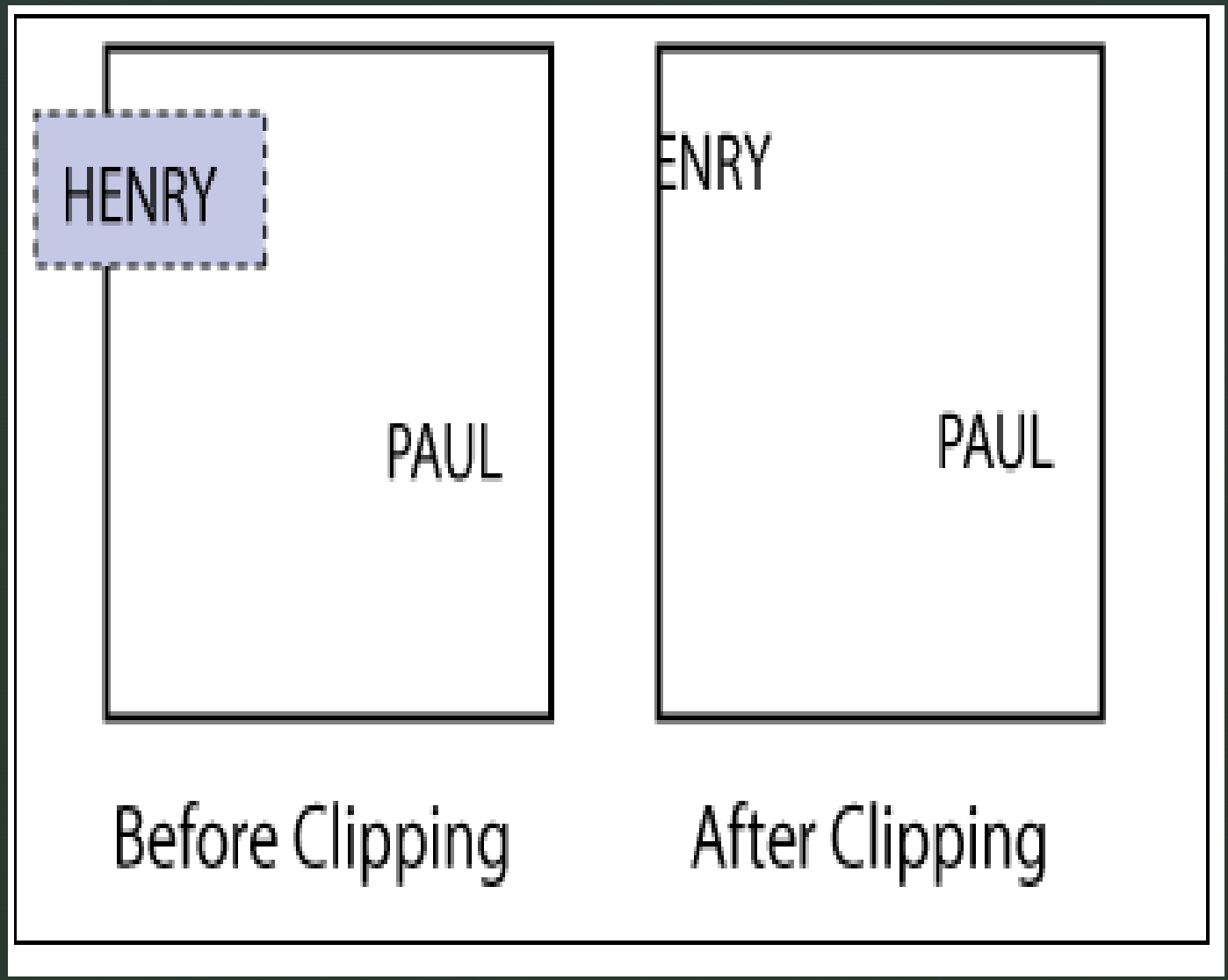
All or none character clipping:

- It is similar to string clipping, but it is based on characters instead of a string. In this method, we compare the character coordinates with window coordinates. There should be following conditions to consider-
- If the character is inside the window, then we will consider it.
- If the character is fully or partially outside the window, then we will remove the character.


PAUL
ADAM
PETER
UL
ADAM
TER
Before Clipping
After Clipping

Text Clipping:

- Itis also known as **“Individual character clipping”** or “**Bitmap character clipping**.”In this method,we consider only those characters that are fully inside the view pane (window). If some portion of the character is outside the view pane then-
- We will remove those portion that is entirely outside the window.
- If any character is lying on the view pane boundary, then we will remove those portion that is outside the window boundary


HENRY
PAUL
ENRY
PAUL
Before Clipping
After Clipping

Polygon Clipping in Computer Graphics

“A Polygon can be described as the enclosed collection or group of the lines.”

In a polygon, all lines are connected. Lines can be a combination of edges and vertices, which together form a polygon. A polygon refers to a two-dimensional architecture made up of a number of straight lines.

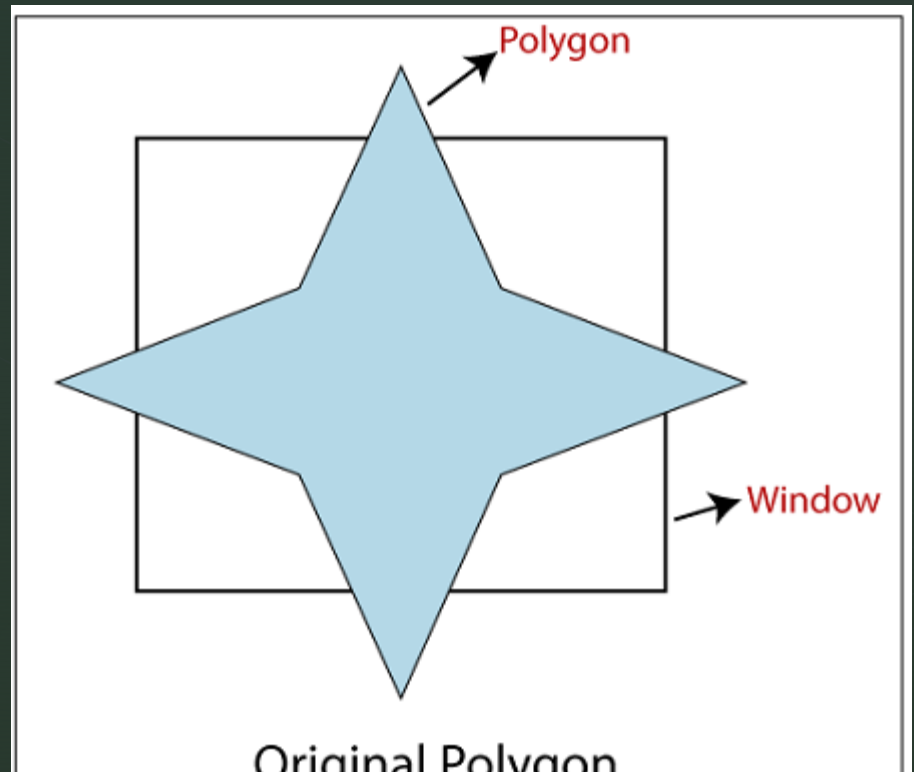
Some Examples of the polygon:

Triangles
Pentagons
Hexagons
Quadrilaterals

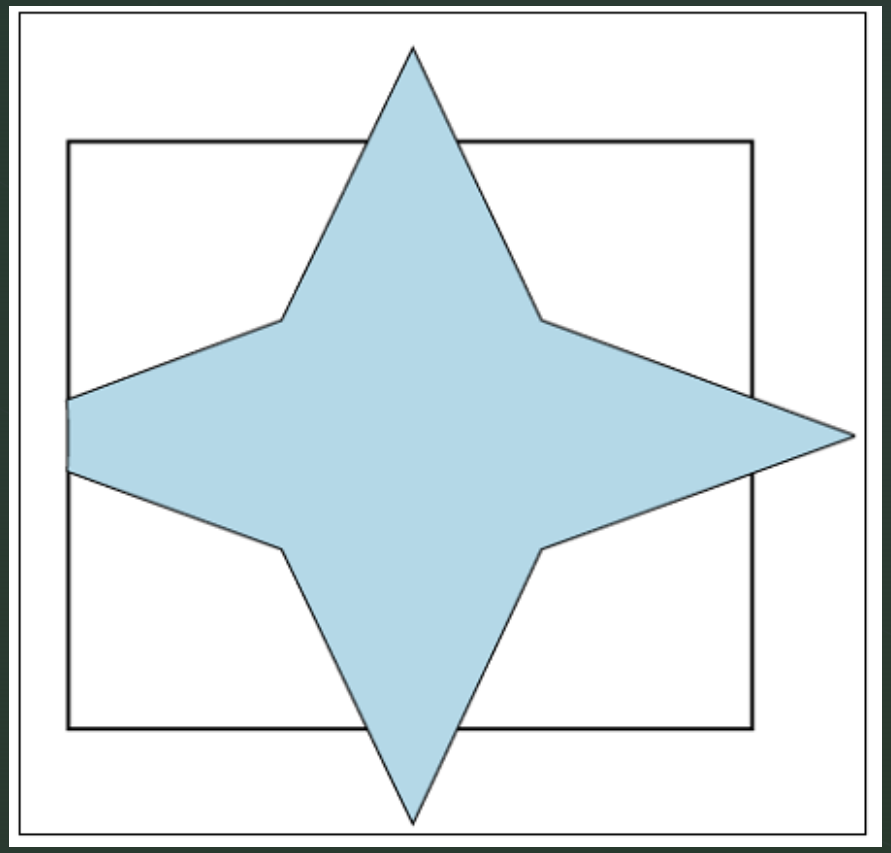
The polygon’s name defines how many sides the architecture contains.

Sutherland-Hodgeman polygon clipping algorithm

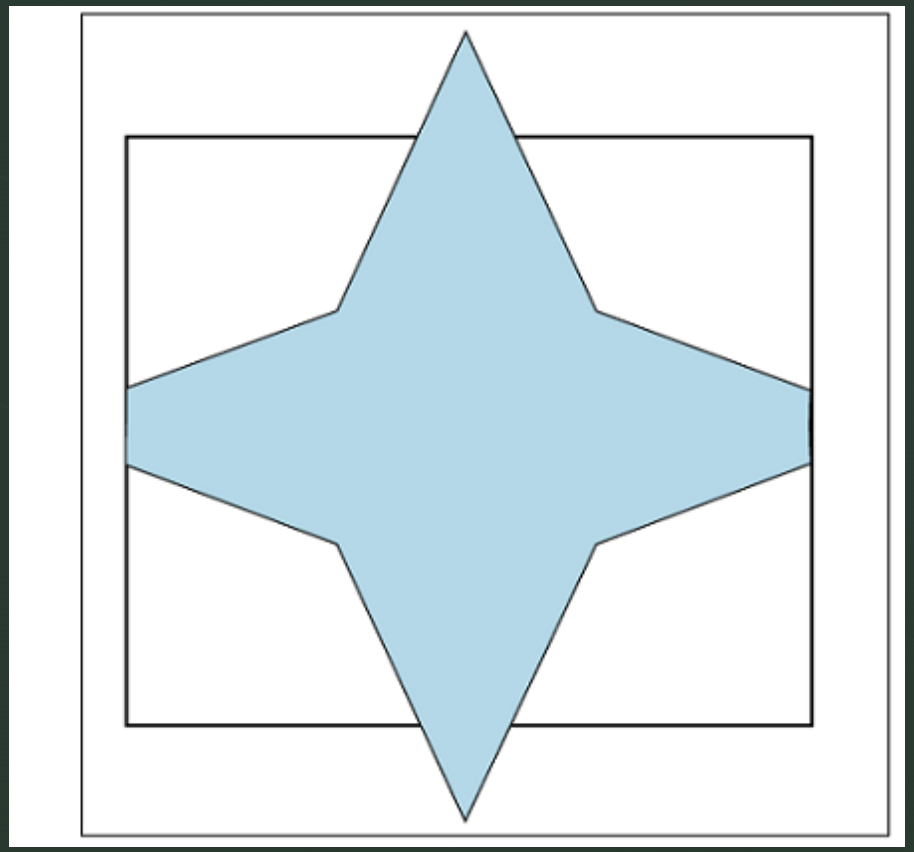
- The polygon clipping algorithm deals with four different clipping cases. The output of each case is input for the next case.
- **Case1) Left clip:** In the left side polygon clipping, we only remove the left part of the polygon, which is outside the window. We only save the portion which is inside the window.



Case2) Right clip: In the right-side polygon clipping, we only remove the right part of the polygon, which is outside the window. We only save the portion which is inside the window.



Case3) Top clip: On the top side polygon clipping, we only remove the top part of the polygon, which is outside the window. We only save the portion which is inside the window.



Case4) Bottom clip: In the bottom side polygon clipping, we only remove the bottom part of the polygon, which is outside the window. We only save the portion which is inside the window.