Difference between the CUI and GUI application without respect to output content to the end user of application.

[1] CUI:
- (a) We can write to any part of character display.
- (b) We don't have to worry about content disappearing mysteriously.
- (c) We can discard the information needed to re-create the screen display.

[2] GUI:
- (a) A programmer can paint only on the client area of the application window.

- (b) The content on the window is not guaranteed to be there until it is overwritten. Putting in other words, it is possible that events other than explicitly over-writing the contents of the window may cause the content to disappear.
  e.g.
  - (i) End user overlaps application's window by some other window.
  - (ii) The window is minimized / resized by the end user.

- (c) In the events such as (b)-(i) and (b)-(ii) the GDI does not make any attempt the save such a content but it will simply discard it and when that content is needed again then the GDI asks the application to re-generate the content.

  As the windows is message driven system, it communicates to the application of any event by posting the message on the message queue or by sending an appropriate event.

  If the GDI wants to communicate to the application that the application needs to re-generate part of it's window's client area or then it does so by positing WM_PAINT message on the message queue of the application.

[3] The WM_PAINT message:
- (a) The first call to WM_PAINT: When an application calls an UpdateWindow() function after ShowWindow(), the first WM_PAINT message is sent (not posted) and the client area of a window is painted for the first time. Thereafter, the application should always be ready for WM_PAINT message (mostly posted, sometimes sent)

- (b) The scenarios when WM_PAINT message is posted/sent to the application
  - (i) A previously hidden area of the window is brought into view when a user moves a window or uncovers a window.
  - (ii) The user resizes the window (if the window class style has CS_HREDRAW or CS_VREDRAW bits set)
  - (iii) The program uses the ScrollWindow() or ScrollDC() function to scroll the part of its client area
  - (iv) The program uses the InvalidateRect() or InvalidateRgn() function to explicitly generate WM_PAINT message.

- (c) The scenarios where the GDI tries to save the client area of the window but its not always successful (WM_PAINT may get sent)
  - (i) The end user removes a dialog box or message box that overlaying the part of the window.
  - (ii) A menu is pulled down and then released.
  - (iii) A tool tip is displayed.

- (d) The scenarios where the GDI always saves the client area of the window.
  - (i) The mouse cursor is moved across the client area.
  - (ii) An icon is dragged across the client area.