Module 4

A window is an area of the screen, usually rectangular in shape, defined by a border that contains a particular view of some area of the computer or some portion of a person's dialog with the computer. It can be moved and rendered independently on the screen. A window may be small, containing a short message or a single field, or it may be large, consuming most or all of the available display space. A display may contain one, two, or more windows within its boundaries.

4.1 WINDOW CHARACTERISTICS

- A window possesses the following characteristics:
  1. A name or title that allows the window to be identified.
  2. A size in height and width (which can vary).
  3. A state, accessible or active, or not accessible. The contents of active windows only can be altered.
  4. Visibility - the portion of the window that can be seen. A window may be partially or fully hidden behind another window, or the information within a window may extend beyond the window's display area.
  5. A location, relative to the display boundary.
  6. Presentation, that is, its arrangement in relation to other windows. It may be tiled, overlapping, or cascading.
  7. Management capabilities, methods for manipulation of the window on the screen.
  8. Its highlight, that is, the part that is selected.
  9. The function, task, or application to which it is dedicated.

4.1.1 The Attraction of Windows

- Windows are valuable in the context of tasks or jobs.
- A person performs a variety of tasks, often in a fairly unstructured manner. A person is asked to monitor and manipulate data from a variety of sources, synthesize information, summarize information, and reorganize information.
- Things are rarely completed in a continuous time frame due to external events such as telephone calls, supervisor or customer requests, and deadlines force shifts in emphasis and focus. Tasks start, stop, and start again.
- In single-screen technology, only one screen of information can be viewed at one time, comparing or integrating information from different sources and on different screens often requires extensive use of one's memory.
- To support memory, a person is often forced to write notes or obtain printed copies of screens.

- Switching between tasks is difficult and disruptive, and later returning to a task requires an extensive and costly restructuring of the work environment.
- Windowing allows the display workspace to mirror the desk workspace reducing one's short-term memory load.
- Windows act as external memories that are an extension of one's internal memory.
- Windows also make it much easier to switch between tasks and to maintain one's context, since one does not have to reestablish one's place continually.
- Windows provide access to more information than would normally be available on a single display of the same size.
- Windows are *useful* in the following ways.

1. ***Presentation of Different Levels of Information:*** Information can be examined in increasing levels of detail. A document table of contents can be presented in a window. A chapter or topic selected from this window can be simultaneously displayed in more detail in an adjoining window. Deeper levels are also possible in additional windows.
2. ***Presentation of Multiple Kinds of Information:*** Variable information needed to complete a task can be displayed simultaneously in adjacent windows. An order-processing system window could collect a customer account number in one window and retrieve the customer's name and shipping address in another window. A third window could collect details of the order, after which another window could present factory availability of and shipping dates for the desired items. Significant windows could remain displayed so that details may be modified as needed prior to order completion. Low inventory levels or delayed shipping dates might require changing the order.
3. ***Sequential Presentation of Levels or Kinds of Information:*** Steps to accomplish a task can be sequentially presented through windows. Successive windows are presented until all the required details are collected. Key windows may remain displayed, but others appear and disappear as necessary. This sequential preparation is especially useful if the information-collection process leads down various paths. Example: An insurance application will include different types of coverage. A requested type of coverage might necessitate the collection of specific details about that type of coverage. This information can be entered into a window presented to collect the unique data. The windows disappear after data entry, and additional windows appear when needed.
4. ***Access to Different Sources of Information:*** Independent sources of information may have to be accessed at the same time. This information may reside in different host computers, operating systems, applications, files, or areas of the same file. It may be presented on the screen alongside the problem, greatly facilitating its solution. Example: a writer may have to refer to several parts of

a text being written at the same time. Or, a travel agent may have to compare several travel destinations for a particularly demanding client.

5. ***Combining Multiple Sources of Information:*** Text from several documents may have to be reviewed and combined into one. Relevant information is selected from one window and copied into another.
6. ***Performing More Than One Task:*** More than one task can be performed at one time. While waiting for a long, complex procedure to finish, another can be performed. Tasks of higher priority can interrupt less important ones. The interrupted task can then be resumed without the necessity to "close down" and "restart."
7. ***Reminding:*** Windows can be used to remind the viewer of things likely to be of use in the near future. Examples: menus of choices available, a history of the path followed or the command choices to that point, or the time of an important meeting.
8. ***Monitoring:*** Changes, both internal and external, can be monitored. Data in one window can be modified and its effect on data in another window can be studied. External events, such as changes in stock prices, out of normal range conditions, or system messages can be watched while another major activity is carried out.
9. ***Multiple Representations of the Same Task:*** The same thing can be looked at in several ways—for example, alternate drafts of a speech, different versions of a screen, or different graphical representations of the same data. A maintenance procedure may be presented in the form of textual steps and illustrated graphically at the same time.

4.1.2 Constraints in Window System Design

- A windows user interface requires a great amount of time the users must spend doing things such as pointing at tiny boxes in window borders, resizing windows, moving windows, closing windows, and so forth.
- The problems with windowing systems can be attributed to three factors: historical considerations, hardware limitations, and human limitations.

1. ***Historical Considerations:*** Historically system developers were more interested in solving hardware problems than user considerations. Less research has been done addressing design issues and their impact on the usability of window systems. Therefore, there are few concrete window design guidelines to aid designers. Lack of guidelines makes it difficult to develop acceptable and agreeable window standards. Many companies have developed style guides, they are very general and limited in scope to their products. Standardization is also made more difficult by the complexity

and range of alternatives available to the designer. Without user performance data, it is difficult to compare realistically the different alternatives, and design choices become a matter of preference.

Standardization of the interface is also hindered by other factors. Software developers who are proud of their originality consider standards as a threat to creativity and its perceived monetary rewards. Some companies are wary of standards because they fear that other companies are promoting standards that reflect their own approach. Some companies have threatened, or brought, legal action against anyone who adopts an approach similar to their own.

Result: developers of new systems create another new variation each time they design a product, and users must cope with a new interface each time they encounter a new windowing system.

2. ***Hardware Limitations:*** Screens are not large enough to take full advantage of windowing capabilities. Many users of personal computers expand their windows to cover a full screen. Either seeing all the contents of one window is preferable to seeing small parts of many windows or the operational and visual complexity of multiple windows is not wanted.

   Slower processing speeds and smaller memory sizes of some computers may prevent the use of windows. A drain on the computer's resources may limit feedback and animation capabilities, thereby reducing the system's usability. Poor screen resolution and graphics capability may also prevent effective use of windows by not permitting sharp and realistic drawings and shapes.

3. ***Human Limitations:*** A windowing system, requires the learning and using of more operations. Much practice is needed to master them. These window management operations are placed on top of other system operations, and window management can become an end in itself. This can severely detract (reduce/weaken) from the task at hand. In a study comparing full screens with screens containing overlapping windows, task completion times were longer with the window screens, but the non-window screens generated more user errors. After eliminating screen arrangement time, task solution times were shorter with windows. The results suggest that advantages for windows do exist, but they can be negated by excessive window manipulation requirements.

   It is also suggested that to be truly effective, window manipulation must occur implicitly as a result of user task actions, not as a result of explicit window management actions by the user.

Other Limitations: Window borders consume valuable screen space. Small windows providing access to large amounts of information can lead to excessive, bothersome scrolling.

4.2 COMPONENTS OF A WINDOW

- A typical window may be composed of up many elements.

- Some appear on all windows and others only on certain kinds of windows, or under certain conditions.
- For consistency purposes, these elements should always be located in the same position within a window.
- Most windowing systems provide consistent locations for elements in their own windows.
- Some inconsistencies exist in
  - element locations between different systems
  - some differences in what the elements are named
  - what graphic images or icons are chosen to identify them.
- A primary window is shown in Figure 4.1.

Figure 4.1: Microsoft Windows primary window.

1. **Frame:**
   - A window will have a frame or border, usually rectangular in shape, to define its boundaries and distinguish it from other windows.
   - A border need not be rectangular but this shape is a most preferred shape.
   - Textual materials, which are usually read from left to right, fit most efficiently within this structure.
   - Border comprises a line of variable thickness and color. This variation aids in identifying the type of window being displayed.
   - Windows filling an entire screen may use the screen edge as the border.
   - Resizable window may contain control points for sizing it.
   - If the window cannot be resized, the border coincides with the edge of the window.

2. **Title Bar:**
   - The title bar is the top edge of the window, inside its border and extending its entire width.
   - Title bar also referred to by some platforms as the *caption, caption bar,* or *title area*.

- The title bar contains a descriptive title identifying the purpose or content of the window.
- In Microsoft Windows, the title bar may also possess, at the extreme left and right ends, control buttons for retrieving the system menu and performing window resizing.
- Also serves as a control point for moving the window and as an access point for commands that apply to a window. Example: as an access point, when a user clicks on the title bar using the secondary mouse button, the pop-up or shortcut menu for the window appears. Pressing the Alt-Spacebar key combination also displays the shortcut menu for the window.
- Title bars are included on all primary and secondary windows.
- Microsoft recommends that application commands or other controls should never be placed in the title bar. Doing so may conflict with the special user controls Windows adds for configurations that support multiple languages.

3. Title Bar Icon:

- Located at the left corner of the title bar in a primary window.
- this button is used in Windows to retrieve a pull-down menu of commands that apply to the object in the window.
- It is 16 X 16 version of the icon of the object being viewed.
- When clicked with the secondary mouse button, the commands applying to the object are presented.
- Microsoft suggests that:
  - If the window contains a tool or utility (that is, an application that does not create, load, and save its own data files), a small version of the application's icon should be placed there instead.
  - If the application creates, loads, and saves documents or data files and the window represents the view of one of its files, a small version of the icon that represents its document or data file type should be placed there.
  - Even if the user has not yet saved the file, display the data file icon rather than the application icon, and again display the data file icon after the user saves the file.

4. Window Sizing Buttons:

- Buttons located at the right corner of the title bar and are used to manipulate the size of a window.
- The leftmost button, the *minimize* button - inscribed with a short horizontal line toward the bottom of the button is used to reduce a window to its minimum size, usually an icon.
- It also hides all associated windows.
- The *maximize* button - typically inscribed with a large box enlarges a window to its maximum size, usually the entire screen.

- When a screen is maximized, the *restore* button replaces the maximize button, since the window can no longer be increased in size.
- The restore button typically inscribed with a pair overlapping boxes returns a window to the size it had before a minimize or maximize action was performed.
- A *close* button typically inscribed with an X closes the window.
- Minimize, maximize, and close buttons are shown in Figure 4.1. These command buttons are graphical equivalents to the actions available through the Title Bar icon.
- Sizing buttons are included on primary windows only.
- All buttons on a primary window's title bar must have equivalent commands on the pop-up or shortcut menu for that window.
- use the following guidelines:
    - When a window does not support a command, do not display its command button.
    - The *Close* button always appears as the rightmost button. Leave a gap between it and any other buttons.
    - The *Minimize* button always precedes the *Maximize* button.
    - The *Restore* button always replaces the *Maximize* button or the *Minimize* button when that command is carried out.

5. What's This? Button:

- *What's This?* Button appears on secondary windows and dialog boxes.
- used to invoke the What's This? Windows command to provide contextual Help about objects displayed within a secondary window.
- When provided, it is located in the upper-right corner of the title bar, just to the left of the close button.
- It is inscribed with a question mark, as illustrated in Figure 4.2.

Figure 4.2: What's This? button.

- On a primary window this command is accessed from the Help drop-down menu.
- This command may also be included as a button on a toolbar or as a command on a popup menu for a specific object.

6. Menu Bar:

- used to organize and provide access to actions.

- located horizontally at the top of the window, just below the title bar.
- contains a list of topics or items that, when selected, are displayed on a pull-down menu beneath the choice.
- A system typically provides a default set of menu actions that can be augmented by an application.
- some platforms call the menu bar an *action bar*.
- contents of the menu bar and its pull-downs are determined by the application's functionality and the context in which the user is interacting with it.

7. Status Bar:

- Information of use to the user can be displayed in a designated screen area or areas.
- They may be located at the top of the screen in some platforms and called a *status area*, or at the screen's bottom.
- Microsoft recommends the bottom location and refers to this area as the *status bar*.
- also referred to by other platforms as a *message area* or *message bar*.
- Microsoft Windows suggests using the status bar to display information about the current state of what is being viewed in the window, descriptive messages about a selected menu or toolbar button, or other non-interactive information.
- It may also be used to explain menu and control bar items as the items are highlighted by the user.

8. Scroll Bars:

- additional information that cannot be presented in a window must be provided by scrolling the display's contents through use of a scroll bar.
- A scroll bar is an elongated rectangular container consisting of a scroll area or shaft, a slider box or elevator, and arrows or anchors at each end.
- For vertical scrolling, the scroll bar is positioned at the far right side of the work area, extending its entire length.
- Horizontal scrolling is accomplished through a scroll bar located at the bottom of the work area.

9. Split Box:

- A window can be split into two or more pieces or panes by manipulating a *split box* located above a vertical scroll bar or to the left of a horizontal scroll bar.
- Also referred to as a *split bar*.
- A window can be split into two or more separate viewing areas that are called *panes*.
- Splitting a window permits multiple views of an object.
- A split window allows the user to:

- Examine two parts of a document at the same time.
- Display different, yet simultaneous, views of the same information.

➢ To support the splitting of a window that is not presplit by design, include a split box.

➢ The split box should be just large enough for the user to successfully target it with the pointer; the default size of a size handle, such as the window's sizing border, is a good guideline.

10. Toolbar:

➢ Toolbars (Figure 4.3) are permanently displayed panels or arrays of choices or commands that must be accessed quickly.

➢ Also called command bars.

➢ designed to provide quick access to specific commands or options.

➢ Specialized toolbars are sometimes referred to as ribbons, toolboxes, rulers, or palettes.

➢ Each toolbar band includes a single-grip handle to enable the user to resize or rearrange the toolbars.

➢ When the user moves the pointer over the grip, it changes to a two-headed arrow. When the user drags the grip, the pointer changes to a split move pointer.

➢ To resize the toolbar to its maximum or minimum size, the user clicks the grip.

➢ Toolbars may occupy a fixed position or be movable.

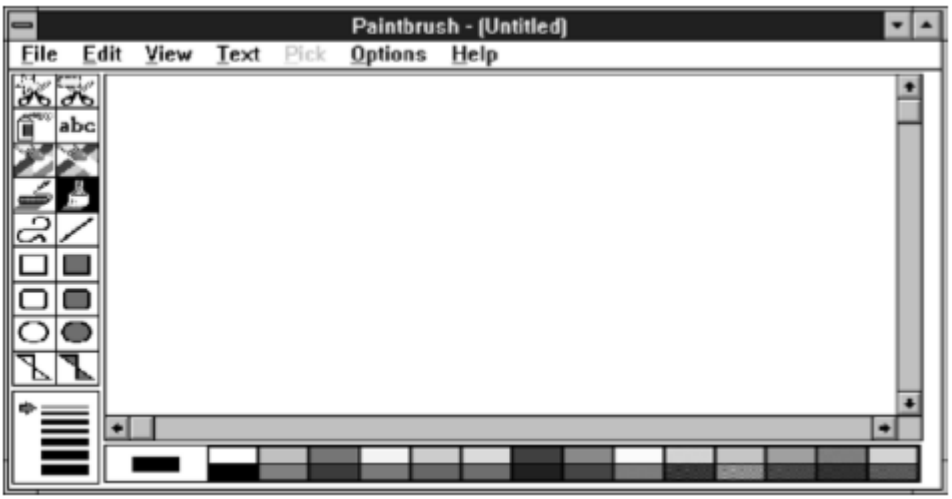

Figure 4.3: Toolbars

11. Command Area:

➢ a command area can be provided for a command to be typed into a screen.

➢ desired location of the command area is at the bottom of the window.

➢ If a horizontal scroll bar is included in the window, position the command area just below it. If a message area is included on the screen, locate the command area just above it.

12. Size Grip:

- A size grip is a Microsoft Windows special handle included in a window to permit it to be resized.
- When the grip is dragged the window resizes.
- Three angled parallel lines in the lower-right corner of a window designate the size grip.
- If the window possesses a status bar, the grip is positioned at the bar's right end. Otherwise, it is located at the bottom of a vertical scroll bar, the right side of a horizontal scroll bar, or the junction point of the two bars. A size grip is shown in the lower-right corner of Figure 4.1.

13. Work Area:

- portion of the screen where the user performs tasks.
- It is the open area inside the window's border and contains relevant peripheral screen components such as the menu bar, scroll bars, or message bars.
- The work area may consist of an open area for typing, or it may contain controls (such as text boxes and list boxes) or customized forms (such as spreadsheets).
- Also referred to as the client area.

4.3 WINDOW PRESENTATION STYLES

- presentation style of a window refers to its spatial relationship to other windows.
- There are two basic styles - tiled or overlapping.

1. Tiled Windows:

- Tiled windows (Figure 4.4) derive their name from common floor or wall tile.
- Tiled windows appear in one plane on the screen and expand or contract to fill up the display surface, as needed.
- systems provide two-dimensional tiled windows, adjustable in both height and width. Less-powerful systems provide one-dimensional windows being adjustable in only one manner (typically height).
- advantages:
  - system usually allocates and positions windows for the user, eliminating the necessity to make positioning decisions.
  - Open windows are always visible, eliminating the possibility of them being lost and forgotten.
  - Every window is always completely visible, eliminating the possibility of information being hidden.
  - They are perceived as less complex than overlapping windows, possibly because there are fewer management operations.
  - They are easier for novice or inexperienced people to learn and use.

- better user performance for tasks where the data requires little window manipulation to complete the task.

➢ disadvantages:

- Only a limited number can be displayed in the screen area available.
- As windows are opened or closed, existing windows change in size.
- As windows change in size or position, the movement can be disconcerting/confusing.
- As the number of displayed windows increases, each window can get very tiny.
- changes in sizes and locations made by the system are difficult to predict.
- The configuration of windows provided by the system may not meet the user's needs.
- They are perceived as crowded and more visually complex because window borders flush against one another, and they fill up the whole screen.
- Crowding is highlighted if borders contain scroll bars or control icons.
- Viewer attention may be drawn to the border, not the data.
- They permit less user control because the system actively manages the windows.

Figure 4.4: Tiled windows

2. Overlapping Windows:

➢ may be placed on top of one another like papers on a desk (Figure 4.5).
➢ They possess a three-dimensional quality, appearing to lie on different planes.
➢ Users can control the location of these windows, as well as the plane in which they appear.
➢ sizes of some types of windows may also be changed.
➢ advantages:

- Visually, their look is three-dimensional, resembling the desktop that is familiar to the user.
- Greater control allows the user to organize the windows to meet his or her needs.
- Windows can maintain larger sizes.
- Windows can maintain consistent sizes.
- Windows can maintain consistent positions.

- Screen space conservation is not a problem, because windows can be placed on top of one another.
- There is less pressure to close or delete windows no longer needed.
- The possibility exists for less visual crowding and complexity. Larger borders can be maintained around window information, and the window is more clearly set off against its background. Windows can also be expanded to fill the entire display.
- They yield better user performance for tasks where the data requires much window manipulation to complete the task.

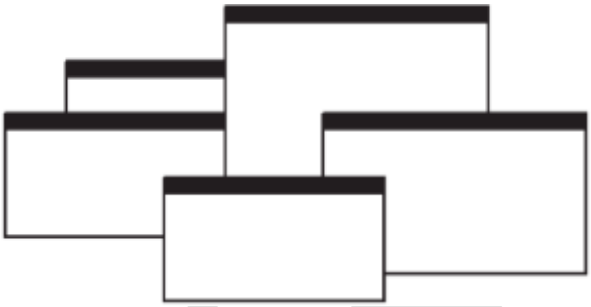

Figure 4.5: Overlapping windows

- Disadvantages:
  - They are operationally much more complex than tiled windows as more control functions require greater user attention and manipulation.
  - Information in windows can be hidden behind other windows.
  - Windows themselves can be lost behind other windows and be presumed not to exist.
  - That overlapping windows represent a three-dimensional space is not always realized by the user.
  - Control freedom increases the possibility for greater visual complexity and crowding. Too many windows, or improper offsetting, can be visually overwhelming.

3. Cascading Windows:

- special type of overlapping window.
- windows automatically arranged in a regular progression.
- Each window is slightly offset from others (Figure 4.6).
- Advantages:
  - No window is ever completely hidden.
  - Bringing any window to the front is easier.
  - It provides simplicity in visual presentation and cleanness.

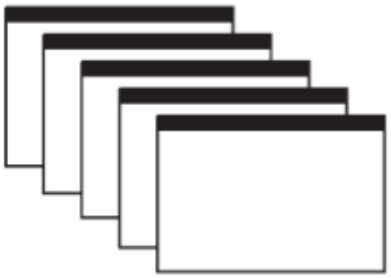

Figure 4.6: Cascading windows

4.3.1 Picking a Presentation Style

1. Tiled windows: Better for single-task activities and data that must be seen simultaneously. Tasks requiring little window manipulation carried out faster using tiled windows. Novice users perform better with tiled windows, regardless of the task.

2. Overlapping windows: Better for situations that necessitate switching between tasks. Tasks requiring much window manipulation performed faster with overlapping windows but only if user window expertise exist. For novice users, tasks requiring much window manipulation carried out faster with tiled windows. Therefore, the advantage to overlapping windows comes only after a certain level of expertise is achieved. Overlapping windows are the preferred presentation scheme.

4.4 TYPES OF WINDOWS

- People's tasks must be structured into a series of windows.
- type of window used depends on the nature and flow of the task.

4.4.1 Primary Window

- first one that appears on a screen when an activity or action is started.
- It is required for every function or application, possessing a menu bar and some basic action controls.
- It should present the framework for the function's commands and data, and provide top-level context for dependent windows.
- also to as the application window or the main window.
- may be referred to as the parent window if one or more child windows exist.
- The primary window is the main focal point of the user's activities and should represent an independent function.

  Avoid dividing an independent function into two or more primary windows, and avoid presenting unrelated functions in a single primary window that tends to confuse people.
- Independent functions should begin in a primary window.

Figure 4.7: Microsoft Windows primary window.

- A primary window should contain constantly used window components such as frequently used menu bar items and controls (for example, control bars) used by dependent windows.
- Also include in a primary window continually updated information such as the date and time.

4.4.2 Secondary window

- Secondary windows are supplemental windows.
- may be dependent upon a primary window or displayed independently of the primary window.
- They structurally resemble a primary window, possessing some of the same action controls (Close button) and possibly a What's This? button.
- A dependent secondary window can only be displayed from a command on the interface of its primary window.
  - It is typically associated with a single data object, and appears on top of the active window when requested.
  - It is movable, and scrollable.
  - If necessary, it uses the primary window's menu bar.
  - Most systems permit the use of multiple secondary windows to complete a task.
  - dependent secondary windows are closed when the primary window closes, and hidden when their primary window is hidden or minimized.
- An independent secondary window can be opened independently of a primary window. Example: a property sheet displayed when the user clicks the Properties command on the menu of a desktop icon.
  - It can typically be closed without regard to the state of any primary window unless there is an obvious relationship to the primary window.

Figure 4.8: Microsoft windows secondary window.

- Uses:
  - used to perform supplemental or subordinate tasks, or tasks that are extended in nature.
  - Frequently and occasionally used window components should also be presented in secondary windows.
  - Microsoft Windows possesses several types of secondary windows called dialog boxes, property sheets, property inspectors, message boxes, palette windows, and pop-up windows.
- Guidelines:
  - A secondary window should typically not appear as an entry on the taskbar.
  - Secondary windows obtain or display supplemental information that is usually related to the objects that appear in a primary window.
  - A secondary window is typically smaller than its associated primary window and smaller than the minimum display resolution.
  - Microsoft recommends not displaying any secondary window larger than 263 dialog units × 263 dialog units.
  - Microsoft defines size and location of user-interface elements in dialog units(DLUs), a device-independent unit of measure.
    - ✓ One horizontal DLU = one-fourth of the average character width for the current system font.
    - ✓ One vertical DLU = one-eighth of the average character height for the current system font.
  - These sizes keep the window from becoming too large to display at most resolutions.
  - provide reasonable space to display supportive information, such as Help information, that applies.

Modal and Modeless:

- A secondary window can be modal or modeless.
- Modal:
  - Most secondary windows will be modal.
  - Modal windows will not permit interaction with another window until the current dialog is completed.
  - It remains displayed until the appropriate action is taken, after which it is removed from the screen.
  - Modal dialog boxes typically request critical information or actions that must be reacted to before the dialog can continue.
  - Modal dialog boxes constrain what the user can do. So, limit their use to situations in which additional information is required to complete a command or when it is important to prevent any further interaction until satisfying a condition.
- Modeless:
  - A modeless dialog box permits the user to engage in parallel dialogs.
  - Switching between the box and its associated window is permitted.
  - Other tasks may be performed while a modeless dialog box is displayed, and it may be left on the screen after a response has been made to it.
  - Actions leading to a modeless dialog box can be canceled, causing the box to be removed from the screen.
  - Use a modeless dialog box when interaction with
    - a primary window or another secondary window must be permitted. Example: during the accessing of the Help function.
    - other windows must be repeated. Example: in a word search operation.

Cascading and Unfolding:

- Access to additional options can be accomplished by inclusion of a command button that opens another secondary window.
- These multiple secondary windows needed to complete a task may be presented in two forms, cascading or expanding.
- Cascading:
  - A cascading window keeps the original window displayed, with the dependent window displayed on top, offset slightly to the right and below the original secondary window.

- A cascade (Figures 4.9 and 4.10) is generally used when advanced options at a lower level in a complex dialog must be presented.
- An indication that the dialog will be cascading is signaled by an ellipsis placed in the command button used to display the additional dialog box.
- restrict the number of cascades to no more than two in a given path to avoid confusion.
- Do not cover information on the upper-level dialog boxes that may have to be referred to, such as box title bars and other critical or relevant information.
- If the cascaded window is independent in its operation, close the secondary window from which it was opened and display only the new window.

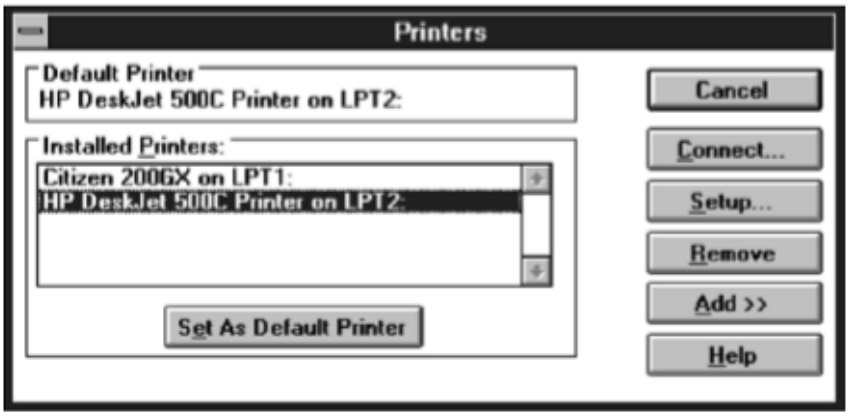

Figure 4.9: Printers secondary window with **Connect...** cascade button

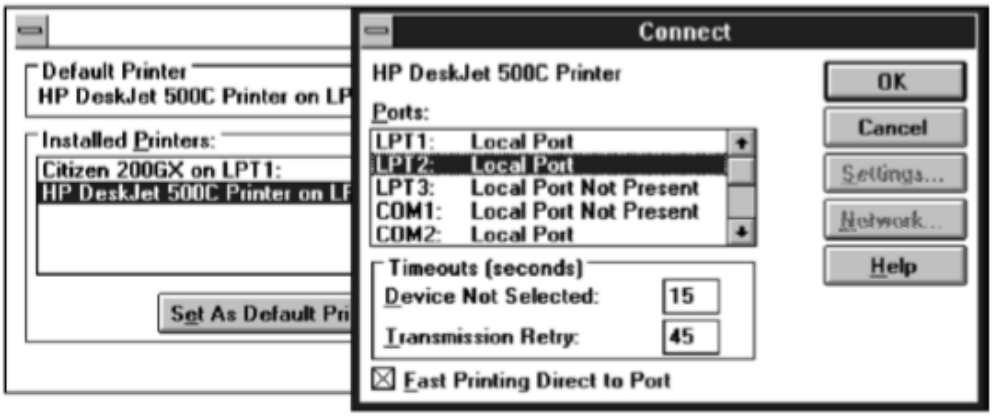

Figure 4.10: Cascading Connect secondary window

- Unfolding:
  - An unfolding secondary window expands to reveal additional options, a form of progressive disclosure.
  - Unfolding windows also called expanding windows, are generally used to provide advanced options at the same level in a complex dialog.

- They are good alternatives when the interface contains a fixed set of options or controls that rarely need to be accessed.
- An unfolding window (Figures 4.11 and 4.12) indication that the dialog will be expanding is signaled by a double arrow (>>) placed in the command button used to display the additional dialog box.
- Expand the box to right, preferably, or downward if screen space constraints exist. As an option, the same button can be used to "refold" the additional part of the window.

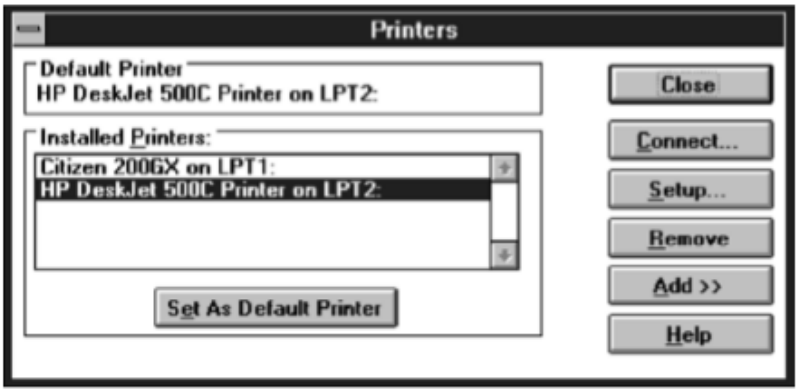

Figure 4.11: printers secondary window with **Add >>** Unfolding button

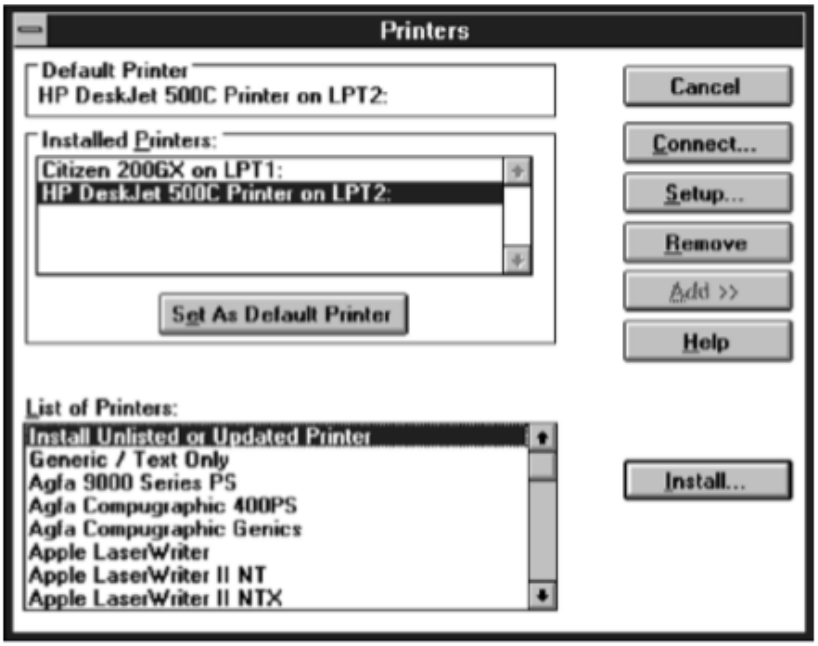

Figure 4.12: Unfolded printers secondary window.

4.4.3 Dialog Boxes

- Dialog boxes are used to extend and complete an interaction within a limited context.
- always displayed from another window, either primary or secondary, or another dialog box.

- They may appear as a result of a command button being activated or a menu choice being selected, or they may be presented automatically by the system when a condition exists that requires the user's attention or additional input.
- They may possess some basic action controls (Close button and possibly a What's This? button), but do not have a menu bar.

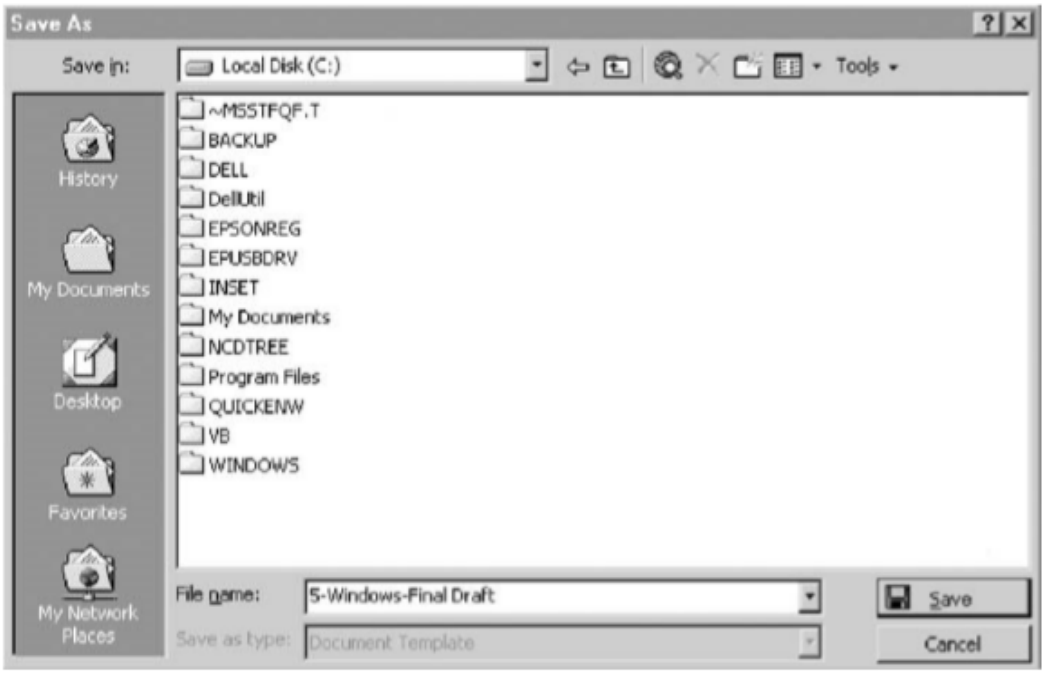

Figure 4.13: Microsoft Windows dialog box.

- Most windowing systems provide standard dialog boxes for common functions. Examples: Open, Save As, and Print.
- Many platforms also recommend a set of standard command buttons for use in the various kinds of dialog boxes, such as OK, Cancel, and so on.
- Dialog boxes are of two types - modal and modeless.
- They may also cascade or unfold.
- Uses: Dialog boxes are used for presenting brief amounts of information or to request specific transient actions. Dialog box actions will usually be those that do not occur frequently.
- Command buttons:
  - Dialog boxes commonly include OK and Cancel command buttons.
  - OK and Cancel buttons work best in dialog boxes that allow the user to set the parameters for a particular command.
  - OK is typically defined as the default command button when the dialog box window opens.
  - Other command buttons may also be included in a dialog box.

4.4.4 Property Sheets and Property Inspectors

- The properties of an object in an interface can be displayed in a variety of ways. Example: the image and name of an icon on the desktop reflect specific properties of that object, as do other interface components such as toolbars, status bars, and even scroll bars.
- Secondary windows provide two other techniques for displaying properties - property sheets and property inspectors.

Property Sheets:

1. Use: A property sheet presents an object's complete set of properties in a secondary window. A property sheet is a modeless secondary window that displays the user-accessible properties of an object, properties that may be viewed but not necessarily edited. A single page property sheet is illustrated in Figure 4.14.

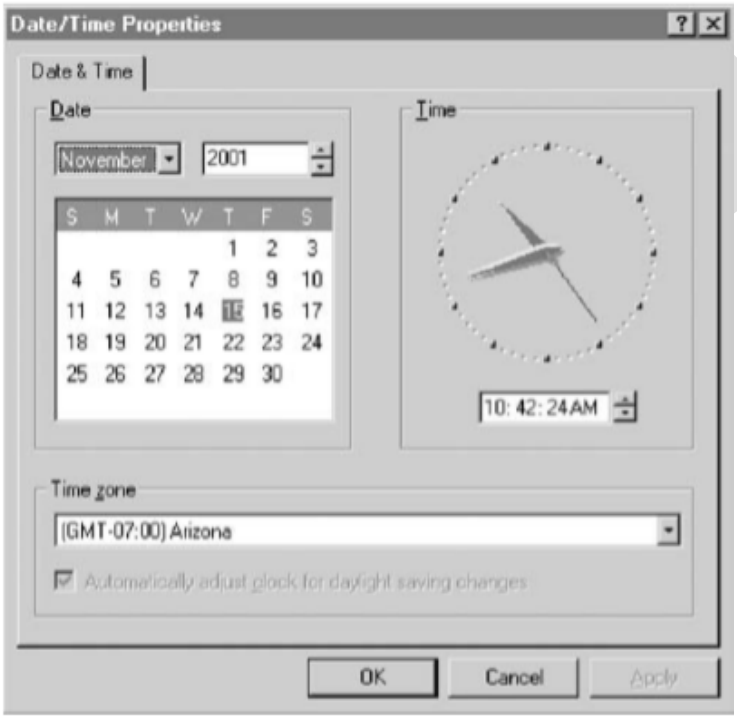

Figure 4.14: Microsoft windows property sheet.

2. Property pages: Because there can be many properties for an object and the object's context, the categorization and grouping of properties within sets may be necessary. A technique for supporting navigation to groups of properties in a property sheet is a tabbed property page, where each set of properties is presented within the window as a separate page. Each page tab is labeled with the name of the set as shown in Figure 4.15. Use tabbed property pages for grouping peer-related property sets.

Figure 4.15: Microsoft windows property sheet tabbed pages.

3. Size: The sizes recommended for property sheets by Microsoft are shown below.

252 DLUs wide x 218 DLUs high
227 DLUs wide x 215 DLUs high
212 DLUs wide x 188 DLUs high

These will create a window smaller than its associated window and smaller than the minimum display resolution.

4. Command buttons: Property sheets typically allow the values for a property to be changed, and then applied. Include the following common command buttons for handling the application of property changes.
   - ✓ For common property sheet transaction buttons, use OK, Cancel, and Apply.
   - ✓ A Reset command button to cancel pending changes without closing the window can also be included.
   - ✓ Other command buttons can be included in property sheets.
   - ✓ Avoid including a Help command button. If a Help button seems necessary, the best solution is to simplify the window.
   - ✓ Command buttons on tabbed property pages should be located outside of the tabbed page but still within the window. Buttons placed on a page imply that the action being

performed applies only to that page. Buttons outside the pages imply the action performed applies to all pages. If the properties are to be applied on a page-by-page basis, however, then place the command and buttons on the property pages, and always in the same location on each page. When the user switches pages without selecting a command button, any property value changes for that page are applied. In these situations, it is useful to prompt the user by displaying a message box that asks whether to apply or discard any changes made.

Property Inspectors:

1. Use: Display only the most common or frequently accessed properties in a property inspector. Properties of an object are displayed by using a dynamic viewer or browser that reflects the properties of the current selection. A property inspector differs from a property sheet. Even when a property sheet window is modeless, the window is typically modal with respect to the object for the properties being displayed. If the user selects another object, the property sheet continues to display the properties of the original object. A property inspector always reflects the current selection.

Figure 4.16: Microsoft windows property inspector

A palette window or a toolbar is used to create a property inspector (Figure 4.16). Use a palette window that the user can also configure. Another control in a property inspector can be used to enable the user to display the properties of various objects in the primary window. For example, as the first control in the property inspector, include a drop-down list box that displays the name of the object being viewed. To view another object's properties within the inspector, the object is selected in the drop-down list box.

2. Dynamic changes: Changes a user makes in a property inspector should be made dynamically. The property value in the selected object should be changed as soon as the user makes the change in the related property control.

- Property inspectors and property sheets are not exclusive interfaces. Both can be included in an interface. The most common or frequently accessed properties can be displayed in a property inspector and the complete set in the property sheet. Multiple property inspectors can also be included, each optimized for managing certain types of objects. An interface's behavior can also be changed

between that of a property sheet and that of a property inspector. A control can be provided that "locks" its view, making it modal to the current object.

4.4.5 Message Boxes

1. Use: A message box (Figure 4.17) is a secondary window that displays a message about a particular situation or condition.

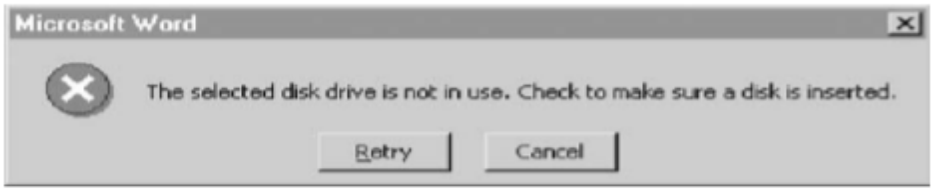

Figure 4.17: Microsoft windows message box.

2. Command buttons: Typically, message boxes contain only command buttons with the appropriate responses or choices offered to the user. The command buttons used should allow the message box interaction to be simple and efficient. Microsoft suggests providing the following:
   - ✓ If a message requires no choices to be made but only acknowledgment, include an OK button and, optionally, a Help button.
   - ✓ If the message requires the user to make a choice, include a command button for each option.
   - ✓ Include OK and Cancel buttons only when the user has the option of continuing or stopping the action.
   - ✓ Use Yes and No buttons when the user must decide how to continue. If these choices are too ambiguous, label the command buttons with the names of specific actions, for example, Save and Delete.

   Command buttons to correct the action that caused the message box to be displayed can be included in a message box. Example: if message box indicates that the user must switch to another application window to take corrective action, a button that opens that application window can be included.
3. Stop: To users, cancel implies that the state of the process or task that started the message is being restored. If you want to interrupt a process and the state cannot be restored, use Stop instead of cancel.
4. Help: A Help button can be included in a message box for messages needing more detail. This allows the message text to be more succinct. If other command buttons are needed, consider the potential increase in complexity that their inclusion will cause.
5. Close box: Enable the title bar Close box only if the message includes a Cancel button. Otherwise, the meaning of the Close operation may be ambiguous.
6. Default: Designate the most frequent or least destructive option as the default command button.

4.4.6 Palette windows

Use: Palette windows are modeless secondary windows that present a set of controls (shown in Figure 4.18). Palette windows are distinguished by their visual appearance, a collection of images, colors or patterns. The title bar for a palette window is shorter and includes only a close button.

Sizing: A palette window can be defined as fixed in size, or resizable by the user. Two techniques should indicate when the window is resizable, changing the pointer image to the size pointer, and placing a size command in the window's shortcut menu. Preserve the window's size and position so the window can be restored if it, or its associated primary window, is closed and then reopened.

Figure 4.18: Microsoft windows palette window

4.4.7 Pop-up windows

- Pop-up windows can be used to display additional information when an abbreviated form of the information is the main presentation technique. (illustrated in Figure 4.19).
- Examples of pop-up windows used to display contextual information are Tool Tips and balloon tips that provide the names for controls in graphical toolbars.
- Popup windows are also used to provide context-sensitive Help information.
- do not contain standard secondary window components such as a title bar and close button.

Figure 4.19: Microsoft windows pop-up window

4.5 WINDOW MANAGEMENT

- Microsoft Windows provides several window management schemes, a single document interface, a multiple-document interface, workbooks and projects.

❖ To choose the right scheme to present an application's collection of related tasks or processes, consider a number of design factors, including: the intended users and their skill level, the application and its objects or tasks, and the most effective use of display space.

1. Single document interface:

- the window interface can be established using a single primary window.
- A single-document window design is sufficient when the object's primary presentation or use is as a single unit, such as a folder or document, even when the object contains different types.
- In a single-document window design, the primary window provides the primary view or work area.
- Secondary windows can be used for supplemental forms of input, and to view information about objects presented in the primary window.
- Use where object and window have a simple, one-to-one relationship.
- Use to support alternate views with a control that allows the view to be changed.
- Use to support simultaneous views by splitting the window into panes.
- Advantages:
  - ✓ Most common usage.
  - ✓ Window manipulation is easier and less confusing.
  - ✓ Data-centered approach.
- Disadvantage: Information is displayed or edited in separate windows.

2. Multiple document interface (MDI):

- MDI may be used when multiple views of an object, or multiple documents, must be looked at simultaneously.
- The purpose of this scheme of windows is to provide multiple views of the same object, to permit comparisons among related objects, and to present multiple parts of an application.
- An MDI interface consists of multiple document windows that are easy to move between, essentially primary windows constrained to appear only within the parent windows boundary (instead of on the desktop).
- These windows may be referred to by a name that describes their contents, such as "Main" in Windows Program Manager.
- When minimized, they are displayed at the bottom of their parent window in iconic form.
- They are also resizable, movable, and scrollable.
- The primary window menu bar content may change dynamically, depending on the MDI window with the focus.

- With MDI, the parent window provides a visual and operational framework for its child windows.
- Child windows typically share the menu bar of the parent window and can also share other parts of the parent's interface, such as a toolbar or status bar. These components can be changed to reflect the commands and attributes of the child window active at that moment.
- Secondary windows displayed as a result of interaction within the MDI parent or child window (dialog boxes, message boxes, or property sheets, for example), typically are not contained or clipped by the parent window.
- These windows should activate and display content according to the conventions for secondary windows associated with a primary window.
- If an MDI document is opened, the MDI parent window opens first, and then the child window for the document opens within it.
- When the parent window is closed, all of its child windows are closed. Where possible, the state of a child window, such as its size and position within the parent window, should be preserved and restored when the user reopens the file.

3. Workbooks:

- A workbook is a scheme for managing a set of views that uses the metaphor of a book or notebook.
- views of objects, in the form of sections, are presented within the workbook's primary window, rather than in individual child windows.
- Tabs are used as a navigational interface to move between different sections.
- Order the tabs to fit the content and organization of the presented information.
- Each tabbed section represents a view of data. One section can be used to list the workbook's table of contents.
- Use:
  - ✓ To manage a set of views of an object.
  - ✓ To optimize quick navigation of multiple views.
  - ✓ For content where the order of the sections is significant.
- Advantages:
  - ✓ Provides a grouping and focus for a set of activities within the larger environment of the desktop.
  - ✓ Conserves screen real estate.
  - ✓ Provides the greater simplicity of the single-document window interface.
  - ✓ Provides greater simplicity by eliminating child window management.
  - ✓ Preserves some management capabilities of the multiple-document interface.

- Disadvantage: Cannot present simultaneous views.

4. **Projects:**
   - A project is similar to a multiple-document interface (MDI) but does not visually contain the child windows.
   - Objects represented by icons contained within it can be opened into primary windows that are peers with the parent window.
   - Opened peer windows in the project do not share the menu bar or other areas contained with the parent window.
   - Each opened peer window within the project must possess its own menu bar and other interface elements.
   - Each peer child window can have its own entry on the taskbar.
   - When a project window is closed, all the peer windows of objects also close.
   - When the project window is opened, the peer windows of the contained objects are restored to their former positions.
   - Peer windows of a project may be restored without the project window itself being restored.
   - Uses: To manage a set of objects that do not necessarily need to be contained. When child windows are not to be constrained.
   - Advantages:
     - ✓ Provides a grouping and focus for a set of activities within the larger environment of the desktop.
     - ✓ Preserves some management capabilities of the multiple document interface.
     - ✓ Provides the greatest flexibility in the placement and arrangement of windows.
   - Disadvantage: Increased complexity due to difficulty in differentiating peer primary windows of the project from windows of other applications.

4.6 ORGANIZING WINDOW FUNCTIONS

- ❖ Information and functions must be presented to people when and where they need them.
- ❖ Proper organization and support of tasks by windows will only be derived through a thorough and clear analysis of user tasks.

4.6.1 Window Organization

- Organize windows to support user tasks as people think in terms of tasks and not functions or applications.
- Support the most common user tasks in the most efficient manner or fewest steps.
- Less frequently performed tasks are candidates for less efficiency or more steps.

- Emphasis on implementation ease considers the needs of the designer than the customer needs.
- Focusing on tasks conforms to the model of how people think.
- Application orientation imposes an unnatural boundary between functions, and lack of cross-team communication seldom yields consistent task procedures.
- Mimicking "what is" will never permit the capabilities of the computer system to be properly exploited.
- Guidelines that recommended usages for the various window types are:
  - Use primary windows to:
    - ✓ begin an interaction and provide a top-level context for dependent windows.
    - ✓ Perform a major interaction.
  - Use secondary windows to:
    - ✓ Extend the interaction.
    - ✓ Obtain or display supplemental information related to the primary window.
  - Use dialog boxes for:
    - ✓ Infrequently used or needed information.
    - ✓ "Nice-to-know" information.

4.6.2 Number of Windows

- Windows are a means to an end, a method of accomplishing something.
- Multiple windows on a display, can be confusing, can increase the load on the human visual system, or may be too small to effectively present what needs to be contained within them.
- The exact number of windows a person can effectively deal with at one time will depend on both the capabilities of the user and the characteristics of the task.
- Some users and situations may permit handling of more than three windows; for other users and situations, three windows may be two too many.
- general rule:
  - ✓ Minimize the number of windows used to accomplish an objective.
  - ✓ Use a single window whenever possible.
  - ✓ Consider the user's task.
  - ✓ Don't clutter up a single window with rarely used information when it can be placed on a second, infrequently used, window.

4.7 WINDOW OPERATIONS

The following are the guidelines for window operations:

1. Active Window:

- the user generally works within a single window at one time though a system supports the display of multiple windows. This window is called the active window.
- Active window may be designated either by the system or the user.
- Many systems make a window active when it is the object of another windowing operation. It is assumed that if users wish to change one aspect of a window's structure, they also wish to change its contents.
- They should also be permitted to move to and make any window active with as few steps as possible. This can be accomplished by simply allowing the user to move the selection cursor to the window's interior and then signal by pressing a key or button.
- For hidden windows, a menu of open windows might be presented from which the user selects a new open window. In some situations, it may be desirable to allow multiple open windows.
- A study compared a single open window with multiple open windows in performing queries. Found that multiple open windows were described by people as more "natural." Performance was slower with multiple open windows.
- If user acceptance is important, multiple open windows might be the better alternative. If speed of task handling is critical, a single active window is more desirable.
- Visually differentiate the active window from other windows. User should be able to quickly identify the active window.
- Active window can be identified by including a contrasting window title bar, border, or background color. An "active" indicator in the window border, which is turned on or off, may also be used.
- A combination of two or more of these visual cues may be used as well.
- Visual cue selected should be of moderate intensity, neither too powerful nor too subtle. Powerful cues will be distracting; subtle cues will be easily overlooked.

2. General Guidelines:

i. Easy to use: Design easy to use and learn window operations. The complexity of a windowing system should not cancel out its potential advantages. Operations must be carefully designed to achieve simplicity. Ideally window manipulations should occur implicitly as a result of the user's task actions, not as a result of explicit, conscious, window management actions.

ii. Minimize number: Minimize the number of window operations needed to achieve a desired effect. Establish the kinds of window operations that people are likely to want, and minimize the number of operations that must be performed to attain these configurations.

iii. Easy navigation: Make navigating between windows easy and efficient. Navigation between windows is the most frequent manipulation activity performed. High-frequency operations should always be easy to do.

iv. Setting up: Make the process of setting up windows easy to remember. Window arrangement (opening, resizing, moving, and so on) is a less frequent activity. Low-frequency operations should always be easy to learn.

v. User-tailorable configurations: In overlapping systems, provide powerful commands for arranging windows in user-tailorable configurations. When an overlapping window system is used, provide easy operations to achieve desired windowing configurations. Specific configurations should be capable of being created, named, and recalled.

3. Opening a window:

- When windows are opened, they are designated as active and positioned in the most forward plane of the screen so that they can be used immediately.
- When opening a window one should not assume a fixed monitor size, but reflect the size of the monitor on which the window will actually be displayed. Window's title bar must be visible.
- To focus attention on the newly opened window, display the screen background behind the window in a neutral or subdued manner.
- When opening windows from an iconic representation, gradually expand the window so that the movement is visible. This will aid association of the icon with the window in the mind of the viewer.
- When a primary window is opened or restored, position it at the top.
- Restore all secondary windows to the states that existed when the primary window was closed.
- When a dependent secondary window is opened, position it on top of its associated primary window.
- If a secondary window has peers, position it on top of its peers.
- Present layered or cascaded windows with any related peer secondary windows.
- When a dependent secondary window is activated, its primary window and related peer windows should also be positioned at the top.
- If more than one object is selected and opened, display each object in a separate window. Designate the last window selected as the active window.
- Always display a window in the same state as when it was last accessed. If the task requires a particular sequence of windows, use a fixed or consistent presentation sequence.
- First opened tiled window consumes the entire screen. Subsequent windows usually positioned by defaults in the system. System positioning of these subsequent windows may not always be consistent

with the user's needs. The system should allow the user to change the default positions, or provide a way for the user to move and resize the system-provided windows easily.

4. Sizing windows:

- Larger windows have these advantages:
  - They permit displaying of more information.
  - They facilitate learning: Data relationships and groupings are more obvious.
  - Less window manipulation requirements exist.
  - Breadth is preferred to depth (based on menu research).
  - More efficient data validation and data correction can be performed.
- Disadvantages include:
  - Longer pointer movements are required.
  - Windows are more crowded.
  - More visual scanning is required.
  - Other windows more easily obscure parts of the window.
  - It is not as easy to hide inappropriate data.
- Always provide large enough windows to present all the relevant and expected information for the task.
- Never hide important or critical information, and minimize the need for scrolling.
- Very small windows requiring a significant amount of scrolling appear to increase decision-making time.
- To avoid scrolling, consider using unfolding dialog boxes, cascading windows, or a tab control.
- Avoid making a window's default size the full size of the display. Doing so leads to any underlying windows being completely hidden from the user's view.
- The option to maximize primary and secondary windows always exists.
- If through analysis and design, a window appears to be too large, determine whether:
  - ✓ all the information is needed
  - ✓ all the information is related
- Important, critical, or frequently used information must be maintained on a screen. If information exists that is needed or used less frequently, place it on another window or dialog box.
- Information included on a screen that is not related to the task being performed should be located with the information to which it is related.

- As a last resort, consider shortening some window control captions or other included window text to achieve a proper fit.
- Procedural text in window sizes of 6, 12, and 24 lines when evaluated found that the fastest and most accurate completion occurred with the 12-line window. The retrieval of alphanumeric information when compared in 7-, 13-, and 19-line windows showed that a seven-line window was found to be more than adequate.

5. Window Placement:

i. Considerations: In placing a window on the display, consider how the window is used in relation to other windows, the overall dimensions of the display, and the reason that the window is being presented.

ii. General: First, locate the window such that it is fully visible. If the window is being restored, locate it where it last appeared. If the window is new and the location has not yet been established, place it at the point of the viewer's attention - usually the location of the pointer or cursor. Also, place the window in a position where it will be convenient to navigate to, and where it will not obscure important underlying screen information. Preferred positions are essentially below and right. The suggested order of placement is below right, below, right, top right, below left, top, left, top left. In a multiple-monitor configuration, display the secondary window on the same monitor as its primary window. If none of these situations applies, horizontally center a secondary window within the primary window, just below the title bar, menu bar, and any docked toolbars. Give each additional window its own unique location. A cascading presentation, from upper left to lower right is recommended. If the user then moves the window, display it at this new location the next time the user opens the window, adjusted as necessary for the current display configuration. Do not let the user move a window to a position where it cannot be easily repositioned.

iii. Dialog boxes: If a dialog box relates to the entire system, center it on display, keeping key information on an underlying window visible. If one dialog box calls another, make the new one movable whenever possible.

6. Window separation:

Component separation is critical in a graphics environment because of the spatial layering that can occur. All windows must be clearly set off from the underlying screen or windows. The demarcation must be crisp and visually pleasing. A solid single-line border is recommended for this purpose. Also provide a window background that sets the window off well against the overall screen background. If color is used, exercise caution and choose compatible colors. Another alternative is to use for the window a

lighter shade of the color used for the screen background. Changes in the density of shades are often more visually pleasing than different colors. To emphasize the three-dimensional aspects of graphic windows, incorporate a drop shadow beneath each window.

7. **Moving a window:**
   - indicate that the move operation has been successfully selected and that the move may begin.
   - should be indicated to the user by changing the pointer's shape. This provides feedback indicating that it is safe to begin the move operation and avoids false starts.
   - Ideally, the entire window should move along with the pointer. If the entire window cannot be moved, move the window outline while leaving the full window displayed on the screen.
   - Displaying only the window's outline during the move operation, and not the window itself, may make it harder for the user to decide when the window has been repositioned correctly.
   - It may be necessary for a window to be moved without being active. This should be possible. The action of moving the window may implicitly activate it.

8. **Resizing a window:**
   - Make primary windows resizable unless the information displayed in the window is fixed or cannot be scaled to provide the user with more information. (For example, a calculator window.)
   - An indication that the resize operation has been successfully selected, and that the move may begin, should be indicated to the user by changing the pointer's shape.
   - This provides the necessary feedback that it is safe to begin the resizing and will avoid false starts.
   - The simplest operation for the user, conceptually, is always to resize from the lower-right corner and anchor the window in the upper-left corner.
   - Flexibility can be provided by permitting resizing to occur from any point on the border (the anchor is always opposite the pulling point), but conceptually this is more complex.
   - Some people may have difficulty predicting which window sides or corners will be resized from specific pulling points.
   - Ideally, the entire window should move along with the pointer. If the entire window cannot be moved, move the window outline while leaving the full window displayed on the screen. Displaying only the window's outline during the move operation, and not the window itself, may make it harder for the user to decide when the window has been repositioned correctly.
   - Effect of resizing operation on the window's contents depends on the application.
   - In enlarging, more data may be displayed, a larger image may be created, or blank space may be added around the image.

- In reducing, less data may be displayed, the image may be made smaller, blank space may be eliminated, or the data may be reformatted.
- If resizing creates a window or image too small for easy use, clip or truncate information arranged in some logical structure, format, or layout. When no layout considerations exist, as is the case for text, format or restructure the displayed information.
- Also consider removing less useful information, if it can be determined.
- When the minimum size has been attained and any additional attempts to reduce window size occur, replace the information with a message that indicates that the minimum size has been reached and that the window must be enlarged to continue working.
- Resizing a window when not active should be possible whenever necessary. The action of moving the window may implicitly activate it.

9. Other operations:

- Maximizing a window increases the size of the window to its largest optimum size. The system default setting for the maximum size is as large as the display. This should be adjustable, as necessary.
- Minimizing a window reduces it to its smallest size.
- Restoring returns a window to its previous size and position after the user has maximized or minimized it.

10. Window shuffling:

- Window shuffling should be easy to perform in as few steps as possible. Two methods are:
  - permit the toggling of the two most recently displayed windows.
  - permit rapid window shuffling and the swapping of the front window and the second or back window.

11. Keyboard control /mouseless operation:

- All window actions should be capable of being performed using the keyboard as well as the mouse. This provides efficient alternative for applications that contain tasks that are primarily keyboard-oriented, for users skilled in touch typing, and for any other situations in which frequent movement between keyboard and mouse may be required.
- The use of mnemonic codes to reflect window mouse actions will greatly aid user learning of the keyboard alternatives.
- To provide user flexibility, all keyboard designations should be capable of being user modified.

12. Closing a window:

- Close a window
  - when the user requests it to be closed
  - when the action required in the window is performed or
  - when the window has no further relevance to the task being performed.
- If the closed window is a primary window, also close its associated secondary windows.
- When a window is closed, its current state, including size and position, should be saved for use when the window is again opened.

4.8 WEB SYSTEMS

Web systems have limited windowing capabilities. The frame concept does provide window-like ability, and JavaScript does provide pop-up windows.

1. Frames:

a. Description: The Web is essentially a single page or a single window entity. A frame is an independent pane of information presented in a Web page, or, as multiple windows. Frames are presented as tiled, with no overlapping capability. The interaction richness, support, and contextual cues provided by overlapping windows are lacking. Frames, then, allow the displaying of multiple documents on a single Web page. These multiple documents can be independently viewed, scrolled, and updated.

b. Proper usage: Frames are useful in situations where portions of the page content are expected to change frequently. The volatile information can be separated from other page content and placed within a frame, thereby requiring only a portion of the page's content to be modified. Frames are also useful for allowing the user to change page content; navigation links can be placed in one frame and the resulting content displayed within another frame. As different links are selected, the content in the related frame changes. Frames more effectively allow users to compare multiple pieces of related information displayed within the different frames.

c. Advantages and disadvantages: Frame disadvantages seem to outweigh the advantages. These disadvantages are being whittled away as Web technology advances. Frames have advantages because of their ability to reduce the user's content comprehension mental load. They are:

   i. They decrease the user's need to jump back and forth between screens, thereby reducing navigation-related cognitive overhead.

   ii. They increase the user's opportunity to request, view, and compare multiple sources of information.

iii. They allow content pages to be developed independently of navigation pages.

The disadvantages mostly cluster around navigational shortcomings, including:

i. The difference between a single Web page and a page with frames is not always obvious to the user.

ii. They suffer some of the shortcomings of tiled screens:
   - ✓ Only a limited number can be displayed in the available screen area.
   - ✓ They are perceived as crowded and more visually complex because frame borders flush against one another and they fill up the whole screen. Crowding is accentuated if the borders contain scroll bars and/or control icons. Viewer attention may be drawn to the border, not the data.

iii. Frames-based pages behave differently from regular Web pages.
   - ✓ Page-printing difficulties and problems can exist.
   - ✓ Page interaction can be clumsy.
   - ✓ URLs cannot be e-mailed to other users.

iv. Frames will not work on older browsers.

d. Guidelines: Consider the following guidelines in using frames:
   - ✓ Use not more than three frames at a time. More frames will shrink each frame's usable area to the point where little space will be available for presenting content. Then, users will not be able to see much and be forced to scroll.
   - ✓ Choose frame sizes based upon the type of information you want to present. Example: Present navigational links in a small frame and the corresponding information in a larger adjacent frame.
   - ✓ Never force people to resize frames to see information. If people feel they must resize frames, the page design is poor.
   - ✓ Do not use more than one scrolling region in frames contained on a page. This may be confusing to many users.
   - ✓ Knowledge related to frame usability will advance with technological advances. Be aware of the latest developments.

2. Pop-Up Windows:

JavaScript pop-up windows pollute screens by their multiplied use. Most frequently used in advertising and are source of great aggravation to almost every user. When a pop-up window begins to appear, most people close them before they are rendered. So, if a pop-up window is used, it may never be completely seen or read by the user. Cautiously use them.

4.9 CHARACTERISTICS OF DEVICE BASED CONTROLS

- Several specific tasks are performed using graphical systems.
  - ✓ To point at an object on the screen.
  - ✓ To select the object or identify it as the focus of attention.
  - ✓ To drag an object across the screen.
  - ✓ To draw something free form on the screen.
  - ✓ To track or follow a moving object. To orient or position an object.
  - ✓ To enter or manipulate data or information.
- Devices vary in how well they perform these actions.
- ***mechanism*** - a direct or indirect pointing device.
  - Direct devices are operated on the screen itself. Examples: light pen, the finger, and voice.
  - Indirect devices are operated in a location other than the screen, most often on the desktop. Examples: mouse, trackball, and keyboard.
  - The psychomotor skills involved in learning to use, and using, a direct device are much simpler than those required for an indirect device.
- ***Direction*** - distance, and speed, what is the relationship between movement of the hand-operated device and the corresponding pointer movement on the screen? Does the pointer movement track control movement exactly or does it not?
  - The mouse achieves a coupled relationship in all three aspects of direction, distance, and speed: The pointer on the screen moves in the direction the mouse is pushed, at the speed the mouse is pushed, and the distance the mouse is pushed (there may be some ratios applied).
  - A trackball does not achieve this relationship in all three aspects. The pointer moves the direction the trackball is turned and the speed the ball is turned, but not the distance the ball is moved because the ball does not move forward or backwards; its socket is stationary.
  - Devices possessing coupled relationships in these three aspects require less psychomotor skill learning than those not possessing a coupled relationship in all three aspects.

4.9.1 Trackball

- trackball is a ball that rotates freely in all directions in its socket. The ball is rotated with one's fingertips, and its direction and speed are tracked and translated into equivalent screen cursor movement. Trackballs are well suited for navigational control, as in video games or exploration of 3-D environments. In these tasks, smooth movement is more important than fine target acquisition. Commonly used with notebook PCs.

- **Advantages**: In terms of direction and speed, a trackball possesses a direct relationship between how it is rolled and how the cursor moves on the screen. The cursor moves in the same direction and speed ratio as the ball is rotated. Many trackballs are mounted on the keyboard itself, permitting the user's hands to remain close to the keys. Trackballs on the keyboard do not require additional desk space, although the keyboard must often be expanded to allow for their inclusion.
- **Disadvantages**: Control movement is in a different plane from the screen, or indirect. The cursor, or pointer, is separated from the control itself—the pointer being on the screen, the control on the keyboard. To effectively use a trackball requires learning the proper psychomotor skills, fine finger movements for accurate pointing, and gross hand movements for moving longer distances. The fine finger movements necessary to use them can be difficult to perform. Over longer periods of use, they can be fatiguing. When paired with keyboard tasks, they require a shift in motor activity from keystrokes to finger/hand movement.

4.9.2 Joystick

- A joystick is a stick or bat-shaped device usually anchored at the bottom. They come in variable sizes, smaller ones being operated by fingers, larger ones requiring the whole hand. The smaller joysticks require fine motor coordination, the larger ones more gross coordination. Force joysticks are immovable, responding to pressure exerted against them. The direction and amount of pressure is translated into pointer movement direction and speed. Movable joysticks can be moved within a dish-shaped area. The direction and distance of the movements create a similar pointer movement on the screen. Some kinds of joysticks permit continuous movements, others only horizontal and vertical movements. Joysticks may also be mounted on the keyboard. Joysticks are also well suited for navigational control where smooth movement is most important.
- **Advantages**: Joysticks typically possess a direct relationship between hand and cursor movement in terms of direction. They do not obscure vision of the screen and, when mounted on the keyboard, do not require additional desk space.
- **Disadvantages**: Indirect devices with control and its result being located in different planes. They require developing a skill to use and can be slow and inaccurate. Use over extended time, they may also be fatiguing. When paired with keyboard tasks, they require a shift in motor activity from keystrokes to finger/hand movement.

4.9.3 Graphic Tablet

- A graphic tablet, also called a touchtablet, is a device with a horizontal surface sensitive to pressure, heat, light, or the blockage of light. It may lie on the desk or may be incorporated on a keyboard,

and it is operated with fingers, light pen, or objects like a pencil or stylus. The screen pointer imitates movement on the tablet.

- **Advantages**: With graphic tablets, a direct relationship exists between touch movements and pointer movements in terms of direction, distance, and speed. The screen mimics the tablet. When used with objects like styluses, light pens, or pencils, the operational angle, horizontal, is more comfortable than those vertically oriented.
- **Disadvantages**: Tablets are also indirect controls, creating coordination problems. To use them requires moving one's hand from the keyboard and, if using another device, picking it up. If the finger is the tablet-activation object, accuracy with small objects is difficult. Tablets also require desk space.

4.9.4 Touch Screen

- A touch screen is a screen that consists of a special surface sensitive to finger or stylus touch. Objects on the screen are pointed to and touched to select them.
- **Advantages**. Touch screens possess a direct relationship between hand and pointer movement in terms of direction, distance, and speed. This relationship is direct because the control (finger or stylus) is on the same plane as the pointer. It does not require any additional desk space.
- **Disadvantages**. They are fatiguing to use over an extended period of time. If a finger is the touch mechanism, it may obscure part of the screen and be too large to be accurate with small objects. A stylus is usually more accurate than the finger. Fingers may also soil the screen, and a stylus may damage it. Both finger and stylus require moving a hand from the keyboard, and if a stylus is used, it must also be picked up.
- **Guidelines**: When using touch screens, larger screen objects should always be provided to foster accuracy in use. Objects should be 3/4″square at a minimum and separated by at least 1/8″. Visual, and perhaps auditory, feedback should be provided in response to activation. When the consequences of selection are destructive, require a confirmation to avoid inadvertent selection. Touch screen devices placed in public places, for use by the general public should possess an instructional invitation to begin their use.

  In touch screens that allow placement of a finger on the screen without item selection, selection is accomplished by lifting the finger off the screen. This may allow more accurate item selection. Another method involves placing a cursor on the screen directly above one's finger and moving the cursor as the finger is moved. The cursor permits better target visibility, as well as the detection of smaller targets.

4.9.5 Light Pen

- A light pen utilizes a touch screen that is sensitive in a specific way to one kind of pen or stylus.
- **Advantages**: Light pens possess a direct relationship between hand and pointer movement in terms of direction, distance, and speed, and are also classified as direct pointing devices because the control (pen or stylus) is on the same plane as the pointer. It does not require any additional desk space, except for a place for the pen to rest. A light pen is usually more accurate than the finger.
- **Disadvantages**: They are also fatiguing to use over an extended period of time. Light pens require moving a hand from the keyboard to pick up and use.

4.9.6 Voice

- Automatic speech recognition technology has been under development for more than a quarter of a century. Its progress has been hindered by the disadvantages listed below.
- **Advantages**. Speech is a simple and direct communication medium. Very useful for people who cannot use a keyboard, or whose hands are otherwise occupied.
- **Disadvantages**. Most speech recognition errors result from the computer speech recognizers' inability to correctly recognize words. People can dictate to a computer at a fairly fast rate, about 105 words per minute. After making the required corrections, the input rate becomes about 25 words per minute when transcribing the input. New users had even lower transcribing rates. Error correction also takes much longer with a speech recognition system. The most commonly used correction methods are: deleting and repeating the last phrase, deleting and repeating a specific word, deleting and selecting a correct word from a list of alternative words, and retyping the selection. Studies have shown that correcting voice recognition errors using a method other than additional voice recognition speeds up the correction process. Noise can hinder the process, and it is very impractical, and disturbing, to try and use it in a very quiet location such as a library.

4.9.7 Mouse

- A mouse is a rectangular or dome-shaped, movable, desktop control containing from one to three buttons used to manipulate objects and information on the screen. The movement of the screen pointer mimics the mouse movement.
- **Advantages**: There is a direct relationship between hand and pointer movement in terms of direction, distance, and speed. The mouse itself contains some basic controls (buttons) useful for manipulating screen objects. The hand position when using the mouse is generally fairly comfortable, and the mouse does not obscure the screen.

- **Disadvantages**: Indirect devices and the control and its result are located in different planes. They require developing a skill to use and, when paired with keyboard tasks, they require movement away from the keyboard and a shift in motor activity from keystrokes to finger/hand movement. The mouse also requires extensive additional desk space and long positioning movements. The mouse comes in a variety of configurations, performs some basic functions, and is operated in several ways.

- **Configurations**: A mouse may possess one, two, or three buttons. Most windowing systems permit operation using all configurations. Buttons are used to perform three functions to be described. When three mouse buttons are not available, the pointer location or keyboard qualifiers must be used to determine the function to be performed. A multibutton mouse permits a more efficient operation, but a person must remember which button to use to perform each function. A multibutton mouse may usually be configured for left- or right hand use.

- **Functions**: Performed by a mouse are Select, Menu, and Adjust. Select function used to manipulate controls, to select alternatives and data, and to select objects that will be affected by actions that follow. Select is the function assigned to a one-button mouse. For a multibutton mouse, it is usually assigned to the leftmost button (assuming a right handed operation). Menu function is typically used to request and display a pop-up menu on a screen. A menu appears when the button is depressed within a particular defined area of the screen. This area may be, for example, the entire screen, within a window, or on a window border. This button eliminates the need for a control icon, which must be pointed at and selected. The user must remember that a menu is available. Adjust function extends or reduces the number of items selected. It is the least used of the three functions and is usually assigned last and given the least prominent location on a mouse.

- **Operations**: 1) movement and positioning of the mouse pointer over the desired screen object. It prepares for a selection or control operation. 2) To press is to hold the button down without releasing it. It identifies the object to be selected. 3) To click is to press and immediately release a button without moving the mouse. This operation typically selects an item or insertion point, operates a control, or activates an inactive window or control. 4) To double-click is to perform two clicks within a predefined time limit without moving the mouse. It is used as a shortcut for common operations such as activating an icon or opening a file. 5) To drag is to press and hold the button down, and then move the pointer in the appropriate direction. It identifies a range of objects or moves or resizes items. 6) To double-drag is to perform two clicks and hold the button down, and then move the pointer in the appropriate direction. It identifies a selection by a larger unit, such as a group of words.

Mouse Usage Guidelines

- Provide a "hot zone" around small or thin objects that might require extremely fine mouse positioning.
- Never use double-clicks or double-drags as the only means of carrying out essential operations.
- Do not use mouse plus keystroke combinations.
- Do not require a person to point at a moving target.

4.9.8 Keyboard

- Christopher Latham Sholes invented the standard typewriter keyboard in 1870. Commonly called the QWERTY layout, Sholes' placement of letters was intended to slow down a typist's keying movements so that the potential for key jams was minimized. From a strictly human-engineering perspective, its layout inadequacies included a dominance of the left hand in making keystrokes, frequent successive keystrokes with the same hand, frequent movement between keyboard key rows, and frequently used letter pairs being placed far from each other. In 1936, August Dvorak created a revised and well-human-engineered keyboard that overcame many of these deficiencies. The advantages of the DVORAK layout, as it came to be called, included a right-hand dominance in keying, much less frequent row changes, and more systematic alternation between the right and left hand. With this new layout, finger travel distances were reduced by at least one order of magnitude. Acceptance of this new keyboard was, and continues, to be slow. Most users have seemed unwilling to invest the time and effort to change.
- **Advantages**. The standard keyboard is familiar, accurate, and does not consume additional desk space. It is useful and efficient for entering or inserting text or alphanumeric data. For tasks requiring heavy text or data entry, shifting the hands between a keyboard and an alternative control, such as a mouse, can be time consuming and inefficient, especially for a touch typist. The keyboard is flexible enough to accept keyed shortcuts, either keyboard accelerators or mnemonic equivalents. Some systems also permit navigation across a screen through use of keyboard keys such as the space bar, arrows, Tab, and Enter. Inefficiencies in using other graphical device-based controls can occur, making it advantageous to use a keyboard. A mouse with a limited number of buttons will require use of the keyboard to accomplish some functions, possibly causing frequent shifting between devices. When operations are being performed on very large screens, the user may also find keyboard window management preferable to the long mouse movements frequently required. Therefore, to compensate for these possible inefficiencies, many windowing systems provide alternative keyboard operations for mouse tasks.

- **Disadvantages**. It requires discrete finger actions to operate instead of the finer positioning movements. As a result, no direct relationship exists in terms of speed and distance between finger or hand movement on the keys and cursor movement on the screen. Depending on the layout of the keyboard cursor control keys, direct-relationship direction problems may also exist, because fingers may not move in the same direction as the cursor. Keyboards will also be slower for non-touch-typists and slower than other controls in pointing tasks.

Keyboard Guidelines

- Provide keyboard accelerators.
  - Assign single keys for frequently performed, small-scale tasks.
  - Use standard platform accelerators.
  - Assign Shift-key combinations for actions that extend or are complementary to the actions of the key or key combination used without the Shift-key.
  - Assign Ctrl-key combinations for:
    - Infrequent actions.
    - Tasks that represent larger-scale versions of the task assigned to the unmodified key.
- Provide keyboard equivalents.
  - Use standard platform equivalents.
  - Use the first letter of the item description.
  - If first letter conflicts exist, use:
    - Another distinctive consonant in the item description.
    - A vowel in the item description.
- Provide window navigation through use of keyboard keys.

QUESTION BANK

1. What is the importance of windows? Explain atleast four components of windows. (8 Marks)
2. What are the different window management schemes? Discuss any two schemes. (8 Marks)
3. Briefly explain the general guidelines followed in designing of window operations. (5 Marks)
4. Describe the guidelines for selecting proper device based controls. (8 Marks)
5. List atleast three advantages and disadvantages of frames in web systems. (3 Marks)
6. Explain the components of window with example. (8 Marks)
7. Discuss briefly about the types of windows with examples (any four). (8 Marks)
8. Describe overlapping windows and tiled windows presentation styles with examples. (8 Marks)
9. Explain the characteristics of touch screen and keyboard. (8 Marks)
10. Explain briefly about the primary window.
11. Explain the various secondary windows. (*any two or three*)

    (*Answer*: dialog boxes, property sheets and inspectors, message boxes, palette windows, pop-up windows)
12. Write short notes on Modal and modeless secondary windows.
13. Write short notes on cascading or unfolding of secondary windows.
14. Explain in detail about property sheets and property inspectors.
15. Explain the frames supported in web systems.
16. Explain the various devices based on direct mechanism of pointing.
17. Explain the various devices based on indirect mechanism of pointing.
18. Mention the various mouse usage guidelines.
19. Explain the different keyboard guidelines.