

# **USER INTERFACE DESIGN**

## **Unit I**

# Definition

- User Interface is the part of software Program that allow users to interact with computer and carry out their task.
- User Interface Design is a subset of a field of study called Human-Computer Interaction.



# HCI

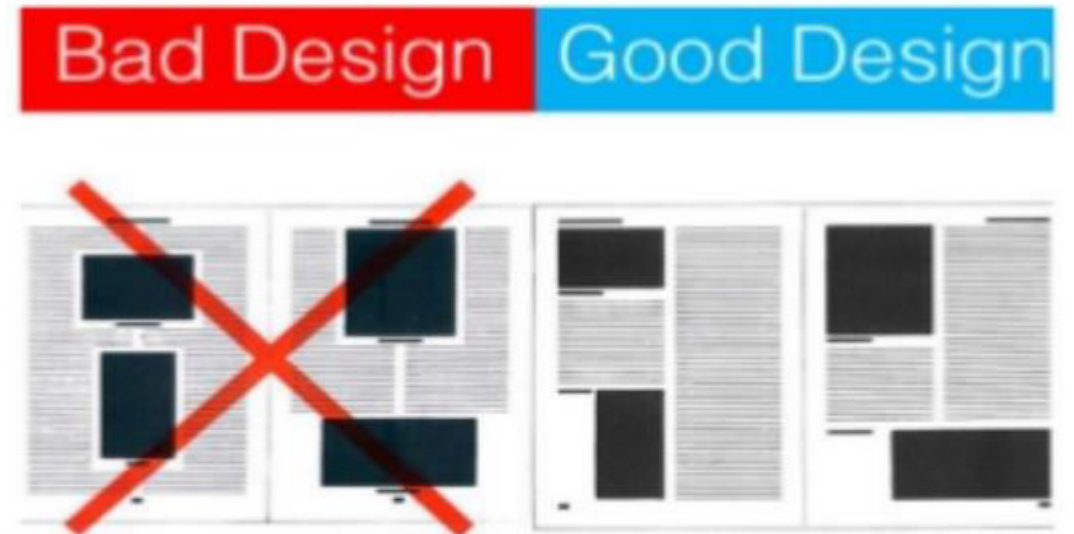
- Human-Computer Interaction is the study about planning and designing of how people and computer works together so that a person's needs are satisfied in the most effective way.
- Key Points to get focus on Designing:
  - What people want and expect?
  - What physical limitations and abilities people possess?

# Components of User Interface

- The user interface has essential two components: Input and Output
- Input is how people communicate his needs to the system using Keyboard or any pointing device
- Output is how the system returns processing result to user through screen or sound.
- **The best interface is one which has proper design with combination of effective input and output mechanisms.**

## Importance of Good Design

- A well-designed interface and screen is important to users.
- It increases the productivity.
- It is their window to view the capabilities of the system and it is also the vehicle through which complex tasks can be performed.



- **Importance of Good Design**

- A screen's layout and appearance affect a person in a variety of ways.
- If they are confusing and inefficient, people will have greater difficulty in doing their jobs and will make more mistakes.
- In spite of today's rich technologies and tools we are unable to provide effective and usable interface design because of lack of time and care.

# Effects of a bad design are:

- People will have greater difficulty in doing their job
- More prone to mistakes
- Chase people away from the system
- Lead to aggravation, frustration and stress

# Benefits of Good Design

- Good design reduces software complexity which makes the software easier to understand and modify.
- Screens are less crowded
- Would be less time consuming, 25 percent less time
- Screen would be 20 percent more productive
- 25 percent fewer errors
- Improve decision making time
- The organization customers benefit because of improved services
- It enables reuse. Good Design makes it easier to reuse code.
- It improves software quality .Good design exposes defects and makes it easier to test the software.





- Training costs are lowered because training time is reduced.
- Support line costs are lowered because fewer assist calls are necessary, and employee satisfaction is increased because aggravation and frustration are reduced.
- Another benefit is, ultimately, that an organization's customers benefit because of the improved service they receive.
- Identifying and resolving problems during the design and development process also has significant economic benefits.

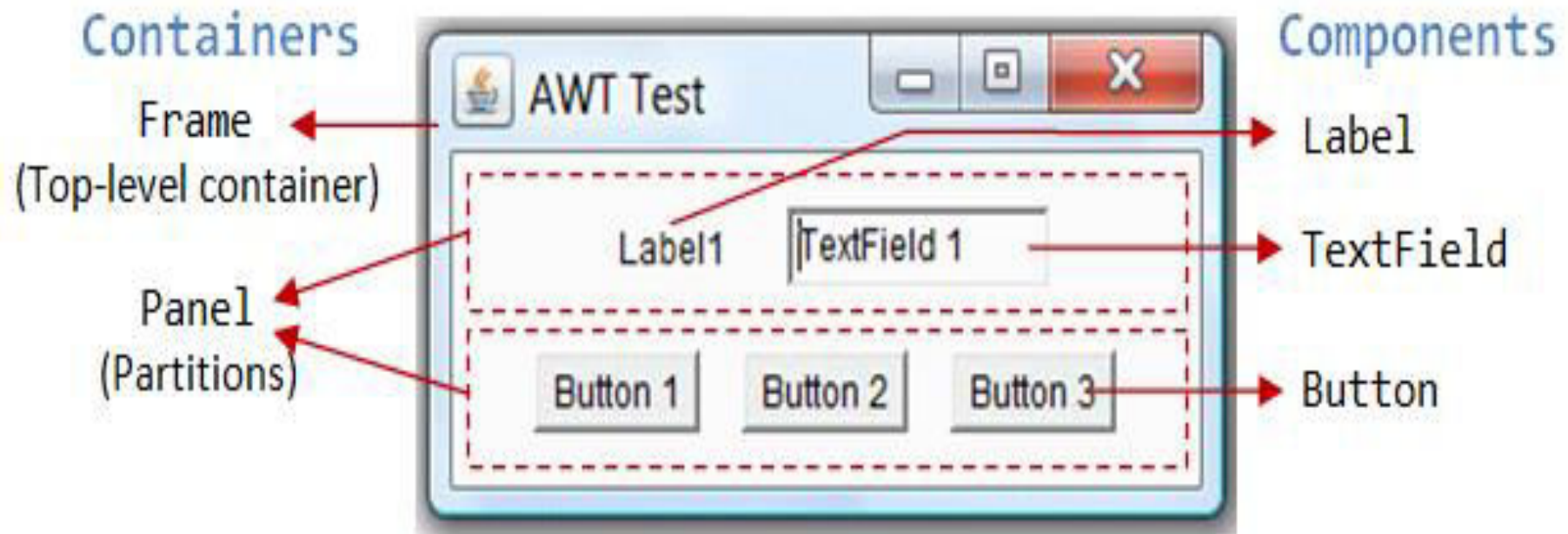
# GUI Definition

- The graphical user interface (GUI) is a form of user interface that allows users to interact with electronic devices through graphical icons and visual indicators .
- GUI Definition. A graphical user interface (GUI) is a human-computer interface (i.e., a way for humans to interact with computers) that uses windows, icons and menus and which can be manipulated by a mouse and keyboard.

## GUI

- A graphical user interface can be defined as a collection of techniques and mechanisms to interact with something.
- In a graphical interface, the primary interaction mechanism is a pointing device of some kind. (Mouse, Joystick, Digital Pen)
- What the user interacts with is a collection of elements referred to as objects. They can be seen, heard, touched, or otherwise perceived.
- Objects are always visible to the user and are used to perform tasks. They are interacted with as entities independent of all other objects.
- People perform operations, called actions, on objects. The operations include accessing and modifying objects by pointing, selecting, and manipulating.

# GUI Objects



# Characteristics of the Graphical User Interface

## ✓ **Sophisticated Visual Presentation**

- Visual presentation is the visual aspect of the interface. It is what people see on the screen.
- The sophistication of a graphical system permits the following:
  - Displaying lines, including drawings and icons.
  - Displaying of a variety of character fonts, including different sizes and styles.
  - Displaying 16 million or more colors on the screen
  - Displays animation, presentation, photographs and motion video.

- The meaningful interface elements visually presented to the user in a graphical system include windows (primary, secondary, or dialog boxes), menus (menu bar, pulldown, pop-up, cascading), icons to represent objects such as programs or files, assorted screen-based controls (text boxes, list boxes, combination boxes, settings, scroll bars, and buttons), and a mouse pointer and cursor.
- The objective is to reflect visually on the screen the real world of the user as realistically, meaningfully, simply, and clearly as possible.

## ✓ **Pick-and-Click Interaction**

- To identify a proposed action is commonly referred to as pick, the signal to perform an action as click.
- The primary mechanism for performing this pick-and-click is most often the mouse and its buttons and the secondary mechanism for performing these selection actions is the keyboard.

## ✓ **Restricted Set of Interface Options**

- The array of alternatives available to the user is what is presented on the screen or what may be retrieved through what is presented on the screen, nothing less, and nothing more. This concept fostered the acronym WYSIWYG (What You See Is What You Get).



## ✓ **Visualization**

- Visualization is a cognitive process that allows people to understand information that is difficult to perceive, because it is either too big or too small.
- Specialized Graphic pictures facilitate visualization.
- The goal is to reproduce a realistic graphical image and conveys the most relevant information.
- Effective visualizations can facilitate mental insights, increase productivity and more accurate use of data.

## ✓ **Object Orientation**

- A graphical system consists of objects and actions. Objects are what people see on the screen as a single unit.
- Objects can be composed of sub objects .For example, an object may be a word document and its subobjects may be a paragraph, sentence, word, and letter.

- Objects are divided into three meaningful classes as Data objects, which present information, container objects to hold other objects and Device objects, represent physical objects in the real world.
- Characteristics of the objects depending upon the relationship exist between them. These relationships are called collections, constraints, composites, and containers.
- **Properties or Attributes of Objects :** Properties are the unique characteristics of an object. Properties help to describe an object and can be changed by users.
- **Actions :** People take actions on objects. They manipulate objects in specific ways (commands) or modify the properties of objects (property or attribute specification).

- The following is a typical property / attribute specification sequence:
  - The user selects an object—for example, several words of text.
  - The user then selects an action to apply to that object, such as the action BOLD.
  - The selected words are made bold and will remain bold until selected and changed again.

### ✓ **Application versus Object or Data Orientation :**

- An application-oriented approach takes an action: object approach, like this:
  - Action> 1. An application is opened (for example, word processing).
  - Object> 2. A file or other object selected (for example, a memo).
- An object-oriented object:action approach does this:
  - Object> 1. An object is chosen (a memo).
  - Action> 2. An application is selected (word processing).

## ✓ **Views**

- Views are ways of looking at an object's information. IBM's SAA CUA describes four kinds of views: composed, contents, settings, and help.

## ✓ **Use of Recognition Memory**

- Continuous visibility of objects and actions encourages to eliminate “out of sight, out of mind” problem

## ✓ **Concurrent Performance of Functions**

- Graphic systems may do two or more things at one time. Multiple programs may run simultaneously.
- It may process background tasks (cooperative multitasking) or preemptive multitasking.
- Data may also be transferred between programs. It may be temporarily stored on a “clipboard” for later transfer or be automatically swapped between programs.

## Concept of Direct Manipulation

- The term used to describe this style of interaction for graphical systems was first used by Shneiderman (1982). He called them “direct manipulation” systems, suggesting that they possess the following characteristics:
- ✓ **The system is portrayed as an extension of the real world:**
  - It is assumed that a person is familiar with the objects and actions, replicate them on different medium of screens.
  - It focused on the data not the applications and tools.
  - A person has power to modify the objects.
  - The physical organization of the system, which most often is unfamiliar, is hidden from view.

## ✓ Continuous visibility of objects and actions:

- Objects are continuously visible.
- Actions to be performed by label buttons replacing complex syntax and commands.
- Reminders of actions to be performed are also obvious. Nelson (1980) described this concept as “virtual reality,” a representation of reality that can be manipulated.
- Hatfield (1981) is credited with calling it “WYSIWYG” (what you see is what you get) and Rutkowski (1982) described it as “transparency,”

# Direct Manipulation - Characteristics

## ✓ **Actions are rapid and incremental with visible display of results :**

- The results of actions are immediately displayed visually on the screen in their new and current form. Auditory feedback may also be provided.
- The impact of a previous action is quickly seen
- The evolution of tasks is continuous and effortless.

## ✓ **Incremental actions are easily reversible:** Finally, actions, if discovered to be incorrect or not desired, can be easily undone.

# Limitations

- All the operation is difficult to conceptualize in the graphical system
- The graphics capability of the system is limited.
- The amount of space available for placing manipulation controls in the window border is limited.
- It is difficult for the people to remember all the necessary operations and actions



# Indirect Manipulation

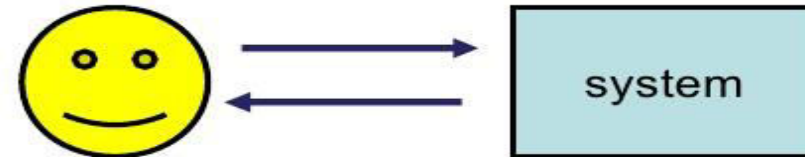
- Style of interaction applied indirectly.
- It substitutes words and text for pull down or pop up menus.
- It substitutes typing for pointing.



# Indirect manipulation

- office – direct manipulation

- user interacts  
with artificial world

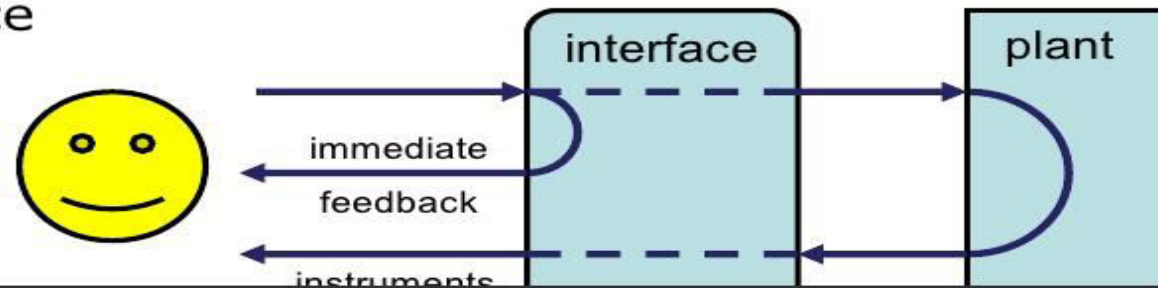


- industrial – indirect manipulation

- user interacts  
*with* real world  
*through* interface

- issues ..

- feedback
- delays



## Graphical system advantages

The success of graphical systems has been attributed to a host of factors. The following have been commonly referenced in literature and endorsed by their advocates as advantages of these systems.

- **Symbols recognized faster than text:** symbols can be recognized faster and more accurately than text. An example of a good classification scheme that speeds up recognition is the icons. These icons allow speedy recognition of the type of message being presented.
- **Faster learning:**

A graphical, pictorial representation aids learning, and symbols can also be easily learned.

- **Faster use and problem solving:** Visual or spatial representation of information has been found to be easier to retain and manipulate and leads to faster and more successful problem solving.
- **Easier remembering:** Because of greater simplicity, it is easier for casual users to retain operational concepts.
- **More natural:** symbolic displays are more natural and advantageous because the human mind has a powerful image memory.
- **Fewer errors:** Reversibility of actions reduces error rates because it is always possible to undo the last step. Error messages are less frequently needed.
- **Increased feeling of control:** The user initiates actions and feels in control. This increases user confidence

- **Predictable system responses:** Predictable system responses also speed learning.
- **More attractive:** Direct-manipulation systems are more entertaining, cleverer, and more appealing.
- **May consume less space:** Icons may take up less space than the equivalent in words but this is not the case always.
- **Replaces national languages:** Icons possess much more universality than text and are much more easily comprehended worldwide.
- **Easily augmented with text displays:** Where graphical design limitations exist, direct-manipulation systems can easily be augmented with text displays. The reverse is not true.
- **Low typing requirements:** Pointing and selection controls, such as the mouse or trackball, eliminate the need for typing skills.

## Graphical system disadvantages

As the coin has two sides, Graphical system also have both advantages and disadvantages. The following are the few disadvantages of Graphical system.

- **Greater design complexity:** Controls and basic alternatives must be chosen from a pile of choices numbering in excess of 50. This design potential may not necessarily result in better design unless proper controls and windows are selected. Poor design can undermine acceptance.
- **Learning still necessary:** The first time one encounters many graphical systems, what to do is not immediately obvious. A severe learning and remembering requirement is imposed on many users because meanings of icons or using pointing device have to be learned.

- **Lack of experimentally-derived design guidelines:** today there is a lack of widely available experimentally-derived design guidelines. Earlier only few studies to aid in making design decisions were performed and available for today now. Consequently, there is too little understanding of how most design aspects relate to productivity and satisfaction.
- **Inconsistencies in technique and terminology:** Many differences in technique, terminology, and look and feel exist among various graphical system providers, and even among successive versions of the same system. So the user has to learn or relearn again while shifting to next terminology.

- **Not always familiar:** Symbolic representations may not be as familiar as words or numbers. Numeric symbols elicit faster responses than graphic symbols in a visual search task.
- **Window manipulation requirements:** Window handling and manipulation times are still excessive and repetitive. This wastes time.
- **Production limitations:** The number of symbols that can be clearly produced using today's technology is still limited. A body of recognizable symbols must be produced that are equally legible and equally recognizable using differing technologies. This is extremely difficult today.
- **Few tested icons exist:** Icons must be researched, designed, tested, and then introduced into the marketplace. The consequences of poor or improper design will be confusion and lower productivity for users.



- **Inefficient for touch typists:** For an experienced touch typist, the keyboard is a very fast and powerful device.
- **Not always the preferred style of interaction:** Not all users prefer a pure iconic interface. User will also prefer alternatives with textual captions.
- **Not always fastest style of interaction:** graphic instructions on an automated bank teller machine were inferior to textual instructions.
- **May consume more screen space:** Not all applications will consume less screen space. A listing of names and telephone numbers in a textual format will be more efficient to scan than a card file.
- **Hardware limitations:** Good design also requires hardware of adequate power, processing speed, screen resolution, and graphic capability.

## Popularity of Graphics

- Graphics revolutionized design and the user interface. Graphics assumes three dimensional look whereas text based system assumes one dimensional look.
- Information can appear or disappear through floating windows and navigation and commands can be done through menu or pull downs or screen controls.
- Increased computer power and the vast improvement in the display enable the user's actions to be reacted to quickly, dynamically, and meaningfully.
- If properly used graphics can reduce mental and perceptual load and increases information transfer between men and machine because of visual comparisons and simplification of the perception of structure.

## The Web User Interface

- Web interface design is essentially the design of navigation and the presentation of information.
- Proper interface design is largely a matter of properly balancing the structure and relationships of menus, content, and other linked documents or graphics. The design goal is to build a hierarchy of menus and pages that feels natural, is well structured, is easy to use, and is truthful.

- The Web is a navigation environment where people move between pages of information, not an application environment. It is also a graphically rich environment.
- Web interface design is difficult for a number of reasons. First, its underlying design language, HTML. Next, browser navigation retreated to the pre-GUI era.
- Web interface design is also more difficult because the main issues concern information architecture and task flow, neither of which is easy to standardize. It is more difficult because of the availability of the various types of multimedia, and the desire of many designers to use something simply because it is available. It is more difficult because users are ill defined, and the user's tools so variable in nature.

# Principles

- **Principle 1: Clarity**
  - User interface must be clear in visual appearance, concept & wording.
- **Principle 2: Aesthetically Pleasing**
  - A design is aesthetically pleasing if it is attractive to the eye
- **Principle 3: Compatibility**
  - User Compatibility
  - Task and Job Compatibility
  - Product Compatibility
- **Principle 4: Comprehensibility**
  - user should know what to look at, what to do, when to do it, where to do it, why to do it & how to do it.
- **Principle 5: Configurability**
- **Principle 6: Consistency**
  - It can reduce requirements for human learning by allowing skills learned in one situation to be transferred to another like it

- **Principle 7:Control**
  - The user must control the interaction & never be interrupted for errors
- **Principle 8:Directness**
  - Tasks are performed by directly selecting an object then selecting an action performed & then seeing the action being performed.
- **Principle 9:Efficiency**
  - Transition between various systems controls should flow easily & freely.
  - Navigation paths should be as short as possible
- **Principle 10:Familiarity**
  - Familiar concepts enable people to get started & become productive quickly.
- **Principle 11: Flexibility**
  - Flexibility is the system's ability to respond to individual differences in people.
  - permitting system customization.
- **Principle 12: Forgiveness**
  - People will make mistakes; a system should be able to tolerate those that are common & unavoidable.
  - A forgiving system keeps people out of trouble.

- **Principle 13: Predictability**

- All actions should lead to results the user expects.
- Current operations should provide clues as to what will come next.

- **Principle 14: Recovery**

- A person should be able to retract any action by issuing an undo command.
- Recovery should be obvious, automatic, easy & natural to perform.

- **Principle 15: Responsiveness**

- A user must be responded quickly.
- Substantial or more informative feedback is most important for the casual or new system user.
- All requests must be acknowledged in some way.

- **Principle 16: Simplicity**

- **Principle 17: Transparency**

- Permit the user to focus on the task or job without concerning the mechanics of the interface.
- Working & reminders of workings inside the computer should be invisible to the user.

- **Principle 18: Trade-offs**

- Final design will be based on a series of trade-offs balancing often-conflicting design principles.
- People's requirements always take precedence over technical requirements.



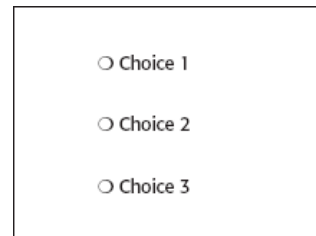
# MENUS

UNIT II

# STRUCTURE OF MENUS:

## Single Menu:

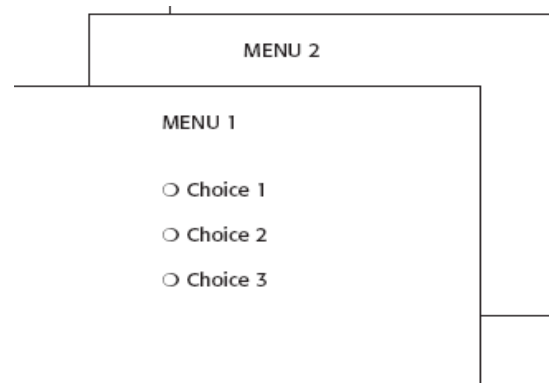
- Simplest form of menu.
- Single screen or window is presented to get the user's input or request an action to be performed.
- Single menu may be iterative if it requires data to be entered and then subject to validity check. If check fails, the menu will represent the user with a message requesting to re-entry valid data.



A diagram illustrating a single menu structure. It consists of a rectangular box containing three radio button options, each preceded by a small circle. The options are listed vertically: "Choice 1", "Choice 2", and "Choice 3".

## Sequential Linear Menus:

- Sequential linear menus are presented on a series of screens possessing only one path.
- The menu screens are presented in a predefined order and generally, their objective is for specifying parameters for entering data.
- A long sequence may become tedious as menu after menu is presented.



## Simultaneous Menu:

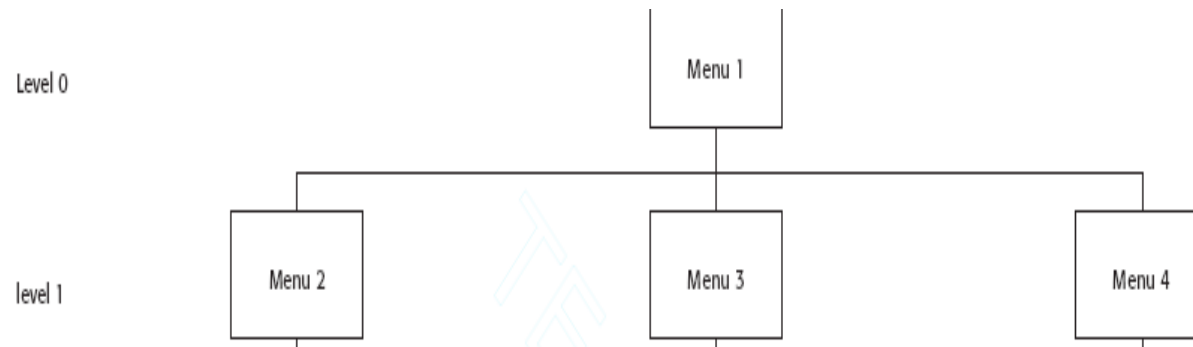
- Instead of being presented on separate screens, all menu options are available simultaneously.
- The menu may be completed in the order desired by the user, choices being skipped and returned to later.
- Problems with simultaneous menus are
  - ✓ For large collections of menu alternatives causes screen clutter easily.
  - ✓ Also be very confusing.

ALTERNATIVE 1	ALTERNATIVE 3
<input type="radio"/> Choice 1	<input type="radio"/> Choice 1
<input type="radio"/> Choice 2	<input type="radio"/> Choice 2
<input type="radio"/> Choice 3	<input type="radio"/> Choice 3
ALTERNATIVE 2	ALTERNATIVE 4
<input type="radio"/> Choice 1	<input type="radio"/> Choice 1
<input type="radio"/> Choice 2	<input type="radio"/> Choice 2
<input type="radio"/> Choice 3	<input type="radio"/> Choice 3
<input type="radio"/> Choice 3	<input type="radio"/> Choice 3

## **Hierarchical Menus:**

- A hierarchical structure results in an increasing refinement of choice as menus are stepped through
- For example, from options to sub-options, from categories to subcategories, from pages to sections to subsections and so on.
- A hierarchical structure can best be represented as an inverse tree, leading to more and more branches as one moves downward through it.
- Common examples of hierarchical design today are found in menu bars with their associated pull-downs.

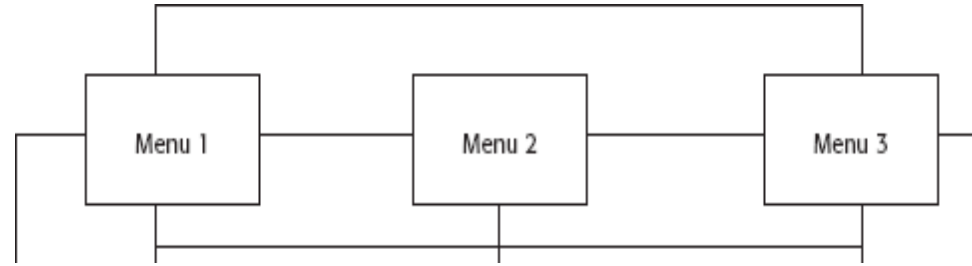
- A disadvantage of a hierarchical scheme is that the defined branching order may not fit the users conception of the task flow.
- If users are not familiar with the hierarchical menu, they are unable to predict what sub-options lie below.
- A particular choice, they may go down wrong paths and find it necessary to go back up the tree to change a choice, or perhaps even return to the top-level menu.



## Connected Menus

- Connected menus are networks of menus all interconnected in some manner.
- Movement through a structure of menus is not restricted to a hierarchical tree, but is permitted between most or all menus in the network.
- A connected menu system may be
  - cyclical with movement permitted in either direction between menus or
  - acyclical with movement permitted in only one direction.
- These menus also vary in connectivity, the extent to which menus are linked by multiple paths.

- **Advantage:** It gives the full control over the navigation flow to user.
- **Disadvantage:** Complexity.



## Event-Trapping Menus:

- Event Trapping menus provide an ever-present background of control over the system's state and parameter while the user working on foreground task.



- Set of simultaneous menus imposed on hierarchical menus.
- These menus can also content based upon the system state or event existing at that moment.
- For example, a paste option in word processing application will only function if there is something in a clipboard to paste.

# FUNCTIONS OF MENUS:

- A menu can be used to perform several functions
  - ✓ to navigate to a new menu
  - ✓ to execute an action or procedure
  - ✓ to display information or
  - ✓ to specify input data or parameters

## **Navigation to a New Menu:**

- Each user selection causes another menu in a hierarchical menu tree to be displayed.
- The purpose of each selection is to steer the user toward an objective or goal.
- Selection errors may lead the user down wrong paths, and cost time and, perhaps, aggravation, but these errors are non-destructive and usually undoable.

## **Execute an Action or Procedure:**

- A user selection directs the computer to implement an action or perform a procedure.
- The action may be something like opening or closing a file, copying text or sending a message.
- Selection errors may or may not have serious consequences depending upon the nature of the action.
- Accidental selection of critical irreversible actions must be prevented in interface design.

## **Displaying Information:**

- The main purpose of selecting a menu choice may simply be to display information.
- The user may be searching for specific information in a database or browsing the Web.
- The user's focus is primarily on the information desired and less on the selection function.
- The content material and the user's interests will determine the paths followed.
- Wrong turns in the process will again cost time and perhaps aggravation, but these errors are non-destructive and usually undoable.

## **Data or Parameter Input:**

- Each selection specifies a piece of input data for the system or provides a parameter value.
- Data or values may be input on a single menu or spread over a hierarchy of menus.
- The user's focus is primarily on the information being provided and less on the selection function.
- Selection errors can easily be corrected if detected by the system.

# CONTENT OF MENUS:

- A menu consists of four elements,
  - ✓ its context,
  - ✓ its title,
  - ✓ its choice descriptions and
  - ✓ its completion instructions.

## **Menu Context:**

- A menu's context provides information to keep the user oriented.
- Feedback is necessary that tells users where they are in a process, what their past choices were, and possibly how much farther they still have to navigate.
- Verbal linkage, spatial linkage, or both may be used to provide navigation feedback.
- Verbal linkage involves providing, on the current menu screen, a listing of choices made on previous menus that have led to this position. It also involves assuring the user that the displayed menu is the menu desired.
- Spatial linkage can be accomplished by graphic methods. Each succeeding menu screen can be displayed overlapping the previous menu screen so a succession of choices can be seen in a single view.



## **Menu Title:**

- A menu's title provides the context for the current set of choices.
- The title must reflect the choice selected on the previously displayed menu.

## **Choice Descriptions:**

- Choice descriptions are the alternatives available to the user.
- These descriptions can range from a mnemonic, numeric, or alphabetized listing of choices to single words or phrases to full sentences or more.

## **Completion Instructions:**

- Completion instructions tell users how to indicate their choices.
- They may include why the user is being asked to make this choice and the impact the choice will have on subsequent processes.
- Explicit instructions may be needed for first time or casual users of a system.
- Experienced users will find overly verbose instructions unnecessary.
- The needs of all system users, and the nature of the system, must again be considered in creating this kind of on-screen guidance.

# FORMATTING OF MENUS:

- The human-computer interface has a history of experimental studies with menus.
- The results of which can be applied to graphical screen and web page menu design and presentation.
- Which follows a series of guidelines for formatting menus.

## **Consistency:**

- Provide consistency with the user's expectations.
- Provide consistency in menu
  - Formatting including organization, presentation, and choice ordering.
  - Phrasing including titles, choice descriptions, and instructions.
  - Choice selection methods.
  - Navigation schemes.

## **Display:**

- If continual or frequent references to menu options are necessary, permanently display the menu in an area of the screen that will not obscure other screen data.
- If only occasional references to menu options are necessary, the menu may be presented on demand.
- Whether to display menu continually or on demand is determined by the menu's frequency of use.
- Always permanently display menus that are frequently referenced.

## **Presentation:**

- Ensure that a menu and its choices are obvious to the user by presenting them with a unique and consistent structure, location, and/or display technique.
- A menu and its choices should be immediately recognizable by the user as being a menu of choices.
- Ensure that other system components do not possess the same visual qualities as menu choices.

## **Organization:**

- Provide a general or main menu.
- Display:
  - All relevant alternatives.
  - Only relevant alternatives.
    - Delete or gray-out inactive choices.
- Match the menu structure to the structure of the task.
  - Organization should reflect the most efficient sequence of steps to accomplish a person's most frequent or most likely goals.
- Minimize number of menu levels within limits of clarity.
  - For Web sites, restrict it to two levels (requiring two mouse clicks) for fastest performance.

- Be conservative in the number of menu choices presented on a screen:
  - Without logical groupings of elements, limit choices to 4 to 8.
  - With logical groupings of elements, limit choices to 18 to 24.
- Provide decreasing direction menus, if sensible.
- Never require menus to be scrolled.
- Provide users with an easy way to restructure a menu according to how work is accomplished.



## **Complexity:**

- Provide both simple and complex menus.
- Simple: a minimal set of actions and menus.
- Complex: a complete set of actions and menus.
- Providing two set of menus will more effectively satisfy the differing needs of the casual and expert user.

## Item Arrangement

- Align alternatives or choices into single columns whenever possible.
  - Orient for top-to-bottom reading.
  - Left-justify descriptions.
- If a horizontal orientation of descriptions must be maintained:
  - Organize for left-to-right reading.

## Ordering:

- Order lists of choices by their natural order.
- For lists associated with numbers, use numeric order.
- For textual lists with a small number of options (seven or less), order by:
  - ✓ Sequence of occurrence.
  - ✓ Frequency of occurrence.
  - ✓ Importance.
  - ✓ Semantic similarity.
- Use alphabetic order for:
  - ✓ Long lists (eight or more options).
  - ✓ Short lists with no obvious pattern or frequency.

- Separate potentially destructive actions from frequently chosen items.
- If option usage changes, do not reorder menus.
- Maintain a consistent ordering of options on all related menus.
  - For variable-length menus, maintain consistent relative positions.
  - For fixed-length menus, maintain consistent absolute positions.

- **Groupings:**
- Create groupings of items that are logical, distinctive, meaningful, and mutually exclusive.
- Categorize them in such a way as to:
  - Maximize the similarity of items within a category.
  - Minimize the similarity of items across categories.
- Present no more than six or seven groupings on a screen.
- Order categorized groupings in a meaningful way.

- If meaningful categories cannot be developed and more than eight options must be displayed on a screen, create arbitrary visual groupings that:
  - Consist of about four or five but never more than seven options.
  - Are of equal size.
- Separate groupings created through either:
  - Wider spacing, or
  - A thin ruled line.
- Provide immediate access to critical or frequently chosen items.

## Line Separators

- Separate vertically arrayed groupings with subtle solid lines.
- Separate vertically arrayed subgroupings with subtle dotted or dashed lines.
- For subgroupings within a category:
  - Left-justify the lines under the first letter of the columnized choice descriptions.
  - Right-justify the lines under the last character of the longest choice description.
- For independent groupings:
  - Extend the line to the left and right menu borders.

# PHRASING THE MENU:

- A menu must communicate to the user information about:
  - ✓ The nature and purpose of the menu itself.
  - ✓ The nature and purpose of each presented choice.
  - ✓ How the proper choice or choices may be selected.



# Menu Titles

- Main menu:
  - ✓ Create a short, simple, clear, and distinctive title, describing the purpose of the entire series of choices.
- Submenus:
  - ✓ Submenu titles must be worded exactly the same as the menu choice previously selected to display them.
- General:
  - ✓ Locate the title at the top of the listing of choices.
  - ✓ Spell out the title fully using either an:
    - Uppercase font.
    - Mixed-case font in the headline style.
  - ✓ Superfluous titles may be omitted.

## **Menu Choice Descriptions :**

- Create meaningful choice descriptions that are familiar, fully spelled out, concise and distinctive.
- Descriptions may be single words, compound words, or multiple words or phrases.
  - Exception: Menu bar items should be a single word (if possible).
- Place the keyword first, usually a verb.
- Use the headline style, capitalizing the first letter of each significant word in the choice description.
- Use task-oriented not data-oriented wording.

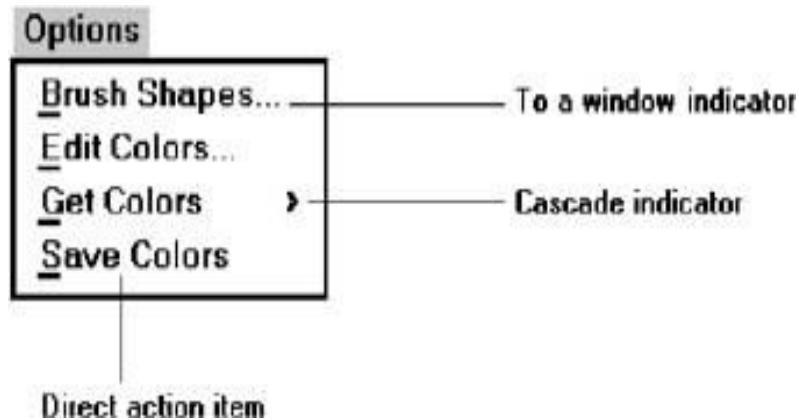
- Use parallel construction.
- A menu choice must never have the same wording as its menu title.
- Identical choices on different menus should be worded identically.
- Choices should not be numbered.
  - Exception: If the listing is numeric in nature, graphic, or a list of varying items, it may be numbered.
- Word choices as commands to the computer.

## Menu Instructions

- For novice or inexperienced users, provide menu completion instructions.
  - ✓ Place the instructions in a position just preceding the part or parts of the menu to which they apply.
    - Left-justify the instruction and indent the related menu choice descriptions a minimum of three spaces to the right.
    - Leave a space line, if possible, between the instructions and the related menu choice descriptions.
  - ✓ Present instructions in a mixed-case font in sentence style.
- For expert users, make these instructions easy to ignore by:
  - ✓ Presenting them in a consistent location.
  - ✓ Displaying them in a unique type style and/or color.

## Intent Indicators

- Cascade indicator:
  - To indicate that selection of an item will lead to a submenu, place a triangle or right-pointing solid arrow following the choice.
  - A cascade indicator must designate every cascaded menu.
- To a window indicator:
  - For choices that result in displaying a window to collect more information, place an ellipsis (. . .) immediately following the choice.
    - Exceptions—do not use when an action:
      - Causes a warning window to be displayed.
      - May or may not lead to a window.
- Direct action items:
  - For choices that directly perform an action, no special indicator should be placed on the menu.



## Keyboard Equivalents

- To facilitate keyboard selection of a menu choice, each menu item should be assigned a keyboard equivalent mnemonic.

- The mnemonic should be the first character of the menu item's description.
  - ✓ If duplication exists in first characters, use another character in the duplicated item's description.
  - ✓ Preferably choose the first succeeding consonant.
- Designate the mnemonic character by underlining it.
- Use industry-standard keyboard access equivalents when they exist.

Style	
<u>N</u> ormal	
<u>B</u> old	Ctrl+B
<u>I</u> talic	Ctrl+I
<u>U</u> nderline	Ctrl+U
<u>O</u> utline	
<u>S</u> hadow	

## Keyboard Accelerators :

- For frequently used items, provide a keyboard accelerator to facilitate keyboard selection.
- The accelerator may be one function key or a combination of keys.
  - ✓ Function key shortcuts are easier to learn than modifier plus letter shortcuts.
- Pressing no more than two keys simultaneously is preferred.
  - ✓ Do not exceed three simultaneous keystrokes.
- Use a plus (+) sign to indicate that two or more keys must be pressed at the same time.
- Separate the accelerator from the item description by three spaces.
- Right-align the key descriptions.



- Do not use accelerators for:
  - ✓ Menu items that have cascaded menus.
  - ✓ Pop-up menus.

# SELECTING MENU CHOICES:

## **Initial Cursor Positioning**

- If one option has a significantly higher probability of selection, position the cursor at that option.
- If repeating the previously selected option has the highest probability of occurrence, position the cursor at this option.
- If no option has a significantly higher probability of selection, position the cursor at the first option.

## Choice Selection:

- **Pointers:**
- Select the choice by directly pointing at it with a mechanical device such as a mouse or trackball pointer, or light pen, or pointing with one's finger.
- If pointing with a mechanical device is the selection method used:
  - ✓ The selectable target area should be at least twice the size of the active area of the pointing device or displayed pointer. In no case should it be less than 6 millimeters square.
  - ✓ Adequate separation must be provided between adjacent target areas.
- If finger pointing is the selection method used:
  - ✓ The touch area must be a minimum of 20 to 30 millimeters square.
  - ✓ The touch area must encompass the entire caption plus one character around it.

- **Keyboard:**
- If moving the cursor to a menu choice:
  - ✓ The up and down arrow keys should move the cursor up or down vertically oriented menu options.
  - ✓ The left and right cursor keys should move the cursor left or right between horizontally oriented menu options.
- If keying a choice identifier value within an entry field:
  - ✓ Locate the entry field at the bottom of the last choice in the array of choices.
  - ✓ Uppercase, lowercase, and mixed -case typed entries should all be acceptable.

- **Selection/execution:**

- Provide separate actions for selecting and executing menu options.
- Indicate the selected choice through either:
  - ✓ Highlighting it with a distinctive display technique.
  - ✓ Modifying the shape of the cursor.
- Permit unselecting choice before execution.
  - ✓ If a menu is multiple choice, permit all options to be selected before execution.

- **Combining techniques:**

- Permit alternative selection techniques, to provide flexibility.

## **Defaults:**

- Provide a default whenever possible.
- Display as bold text.

## **Unavailable Choices:**

- Unavailable choices should be dimmed or “grayed out.”
- Do not add or remove items from a menu unless the user takes explicit action to add or remove them through the application.

## Mark Toggles or Settings:

- Purpose:
  - Use to designate that an item or feature is active or inactive over a relatively long period of time.
  - Use to provide a reminder that an item or feature is active or inactive.
- Guidelines:
  - Position the indicator directly to the left of the option.

Regular	F5
✓ <b>B</b> old	Ctrl+B
✓ <b><i>I</i></b> talic	Ctrl+I
<u>U</u> nderline	Ctrl+U
<b>S</b> uperscript	
<b>S</b> ubscript	
<u>R</u> educe Font	
<u>E</u> nlarge Font	
<u>F</u> onts...	

**Figure 4.11** Mark toggles.

## Toggled Menu Items

- Purpose:
  - Use to designate two opposite commands that are accessed frequently.
  - Use when the menu item displayed will clearly indicate that the opposite condition currently exists.
- Guidelines:
  - Provide a meaningful, fully spelled-out description of the action.
  - Begin with a verb that unambiguously represents the outcome of the command.
  - Use mixed-case letters, with the first letter of each word capitalized.



# KINDS OF GRAPHICAL MENUS:

## 1. Menu Bar:

- Proper usage:
  - To identify and provide access to common and frequently used application actions that takes place in a wide variety of different windows.
  - A menu bar choice by itself should not initiate an action.
- The advantages of menu bars are that they:
  - Are always visible, reminding the user of their existence.
  - Are easy to browse through.
  - Are easy to locate consistently on the screen.
  - Allow for use of keyboard equivalents

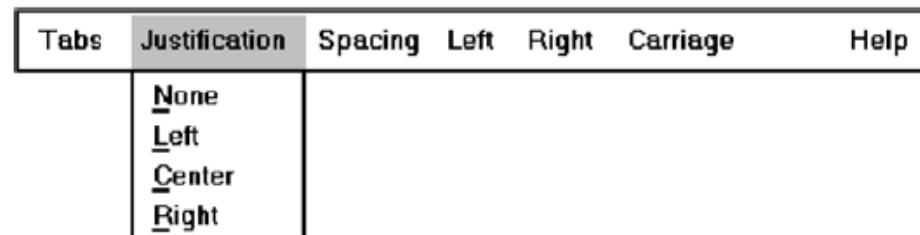
- The disadvantages of menu bars are that:
  - They consume a full row of screen space.
  - They require looking away from the main working area to find.
  - They require moving pointer from the main working area to select.
  - The menu options are smaller than full-size buttons, slowing selection time.
  - Their horizontal orientation is less efficient for scanning.
  - Their horizontal orientation limits number of choices that can be displayed



## 2. Pull-Down Menu

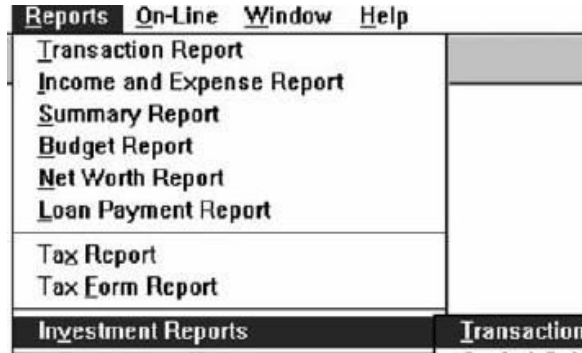
- Proper usage:
  - To initiate frequently used application actions that take place on a wide variety of different windows.
  - A small number of items.
  - Items best represented textually.
  - Items whose content rarely changes.
- The advantages of pull-down menus are:
  - They may be located relatively consistently on the screen.
  - No window space is consumed when they are not used.
  - They are easy to browse through.
  - Their vertical orientation is most efficient for scanning and grouping.

- Their vertical orientation permits more choices to be displayed.
- They allow for display of both keyboard equivalents and accelerators.
- The disadvantages of pull-down menus are:
  - They require searching and selecting from another menu before seeing options.
  - They require looking away from main working area to read.
  - They require moving the pointer out of working area to select (unless using keyboard equivalents).
  - The items are smaller than full-size buttons, slowing selection time.



### 3. Cascading Menus:

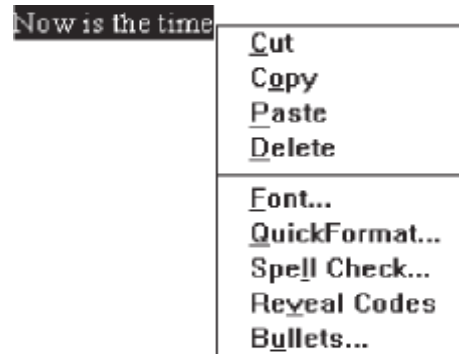
- Proper usage:
  - To reduce the number of choices presented together for selection (reduce menu breadth).
  - When a choice leads to a short, fixed list of single-choice properties.
  - To simplify a higher-level menu.
  - Avoid using for frequent, repetitive commands.
- The advantages of cascading menus are that:
  - The top-level menus are simplified because some choices are hidden.
  - High-level command browsing is easier because subtopics are hidden.
- The disadvantages of cascading menus are:
  - Access to submenu items requires more steps.



## 4. Pop-up Menus

- Use to present alternatives or choices within the context of the task.
- The advantages of pop-up menus are:
  - They appear in the working area.
  - They do not use window space when not displayed.
  - No pointer movement is needed if selected by button.

- The disadvantages of pop-up menus are:
  - Their existence must be learned and remembered.
  - They require a special action to see the menu (mouse click).
  - Items are smaller than full-size buttons, slowing selection time.
  - Their display locations may not be consistent.



## 5. Tear-off Menus

- It may also be called a pushpin, detachable, or roll-up menu.
- Its purpose is to present alternatives or choices to the user that are frequently used and rarely changing in content.
- Follow all relevant guidelines for pull-down menus.
- Advantages/disadvantages.
  - No space is consumed on the screen when the menu is not needed.
  - When needed, it can remain continuously displayed.



## 6. Iconic Menus:

- Use to remind users of the functions, commands, attributes, or application choices available.
- Create icons that:
  - Help enhance recognition and hasten option selection.
  - Are concrete and meaningful.
  - Clearly represent choices



## **Advantages/disadvantages:**

- Pictures help facilitate memory of applications, and their larger size increases speed of selection.
- Pictures consume considerably more screen space than text, and they are difficult to organize for scanning efficiency.
- To create meaningful icons requires special skills and an extended amount of time.
- Iconic menus should be used to designate applications or special functions within an application.
- Icons must be meaningful and clear. They should help enhance recognition and hasten option selection.

# NAVIGATION AND SELECTION:

- General:
  - Permit multiple methods for selecting commands.
- Keyboard equivalents:
  - Assign a mnemonic for each command.
  - A mnemonic should be as meaningful as possible. Use:
    - The first letter of the command, or if duplications exist,
    - The first letter of another word in the command, or
    - Another significant consonant in the command.
  - For standard commands, use mnemonics provided by the tool set.
- Keyboard accelerators:
  - Assign keyboard accelerators for frequently used commands.
  - For standard commands, use keyboard accelerators provided by the tool set.

# WINDOWS

UNIT III

# WINDOW CHARACTERISTICS:

- A window is seen to possess the following characteristics:
  - A **name or title**, allowing it to be identified.
  - A **size**, height and width which can vary.
  - A **state**, accessible or active or not accessible. (Only active windows can have their contents altered.)
  - **Visibility**, the portion 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.)
  - **Presentation**, that is, its arrangement in relation to other windows. It may be tiled, overlapping, or cascading.
  - Its **highlight**, that is, the part that is selected.

# 1. Attraction of Windows:

- Windows seems to be useful in the following ways.
  - ✓ **Presentation of Different Levels of Information.**
    - Table of content in one window simultaneously more detail in adjoining window.
  - ✓ **Presentation of Multiple Kinds of Information.**
    - Variable information needed to complete a task can be displayed simultaneously in adjacent windows.
  - ✓ **Sequential Presentation of Levels or Kinds of Information.**
    - Steps to accomplish a task can be sequentially presented through windows.
    - Key windows may remain displayed, but others appear and disappear as necessary.

✓ **Access to Different Sources of Information.**

- Independent sources of information may be accessed at the same time.

✓ **Combining Multiple Sources of Information.**

- Text from several documents may be reviewed and combined into one.

✓ **Performing More Than One Task.**

- While waiting for a long, complex procedure to finish, another can be performed.
- Tasks of higher priority can interrupt less important ones and then the interrupted tasks can be preceded.

✓ **Reminding.**

- It can be used to provide reminder through messages or popup or menus.

✓ **Multiple Representations of the Same Task.**

- The same task can be represented in two different ways in two windows.

## **2. Constraints in Window System Design:**

- Historically, system developers have been much more interested in solving hardware problems than in user considerations.
- This lack of guidelines makes it difficult to develop acceptable and agreeable window standards.
- The result is that 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.



# COMPONENTS OF A WINDOW:

## **1. Frame:**

- A window will have a frame or border to define its boundaries and distinguish it from other windows.
- Usually rectangular in shape but need not be rectangular.

## **2. Title Bar:**

- The title bar is the top edge of the window, inside its border and extending its entire width.

- Also referred as the *caption*, *caption bar*, or *title area* by some platforms.
- The title bar contains a descriptive title identifying the purpose or content of the window.

### **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.

## 4. Window Sizing Buttons:

- Located at the right corner of the title bar, these buttons 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.
- 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.

## 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:

- The *What's This?* Button, which appears on secondary windows and dialog boxes, is used to invoke the What's This?
- Windows command to provide contextual Help about objects displayed within a secondary window.

## 6. Menu Bar:

- A menu bar is used to organize and provide access to actions. It is located horizontally at the top of the window, just below the title bar.
- A menu bar contains a list of topics or items that, when selected, are displayed on a pull-down menu beneath the choice.

## 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*. It is also referred to by other platforms as a *message area* or *message bar*.

## 8. Scroll Bars

- When all display information cannot be presented in a window, the additional information must be found and made visible.

- This is accomplished 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.

## 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.
- A split box is sometimes referred to as a *split bar*.
- A window can be split into two or more separate viewing areas that are called *panes*.

## 10. Toolbar

- Toolbars are permanently displayed panels or arrays of choices or commands that must be accessed quickly. They are sometimes called *command bars*.
- Toolbars are designed to provide quick access to specific commands or options. Specialized toolbars are sometimes referred to as *ribbons*, *toolboxes*, *rulers*, or *palettes*.

## 11. Command Area

- In situations where it is useful for a command to be typed into a screen, a command area can be provided.
- The desired location of the command area is at the bottom of the window.



## 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, following the same conventions as the sizing border. Three angled parallel lines in the lower-right corner of a window designate the size grip.

## 13. Work Area

- The work area is the 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 also be referred to as the *client area*.

# WINDOW PRESENTATION STYLES:

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

## **1. Tiled Windows:**

- Tiled windows 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.
- Most systems provide two-dimensional tiled windows, adjustable in both height and width.



## **Advantages:**

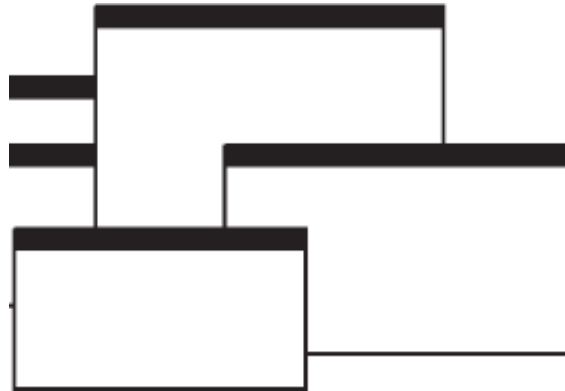
- System allocates and positions windows for the user.
- Every window is always completely visible.
- Eliminating the possibility of them being lost and forgotten.
- Perceived as fewer complexes than overlapping windows.
- Easier for novice or inexperienced people to learn and use.

## **Disadvantages:**

- Limited number can be displayed in the screen area available.
- As windows are opened or closed, existing windows change in size. This can be annoying.
- As number of displayed windows increases, each window can get very tiny.
- Difficult to predict because of change in size and location.
- Configuration by system may not meet the user needs.
- Perceived as crowded and more visually complex.
- Permit less user control.

## 2. Overlapping Windows:

- Overlapping windows may be placed on top of one another like papers on a desk.
- They possess a three-dimensional quality, appearing to lie on different planes.



## **Advantages:**

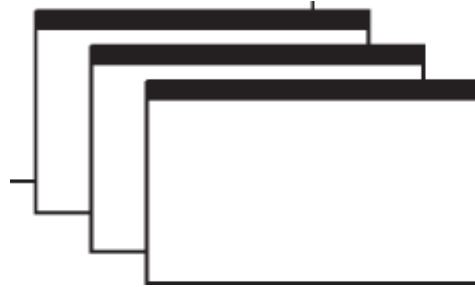
- Visually, their look is three-dimensional.
- Greater control allows the user to organize the windows to meet his or her needs.
- Larger size and Consistent position.
- Screen space conservation is not a problem.
- Less visual crowding and complexity.

## **Disadvantages:**

- Operationally much more complex than tiled windows.
- Information in windows can be hidden behind other windows.

### 3. Cascading Windows:

- A special type of overlapping window has the windows automatically arranged in a regular progression.
- Each window is slightly offset from others, as illustrated in Figure.



#### Advantages:

- No window is ever completely hidden.
- Bringing any window to the front is easier.
- It provides simplicity in visual presentation and cleanness.

## Picking a Presentation Style:

- **Use tiled windows for:**

- ✓ Single-task activities.
- ✓ Data that needs to be seen simultaneously.
- ✓ Tasks requiring little window manipulation.
- ✓ Novice or inexperienced users.

- **Use overlapping windows for:**

- ✓ Switching between tasks.
- ✓ Tasks necessitating a greater amount of window manipulation.
- ✓ Expert or experienced users.
- ✓ Unpredictable display contents.



# TYPES OF WINDOWS:

## 1. Primary Window:



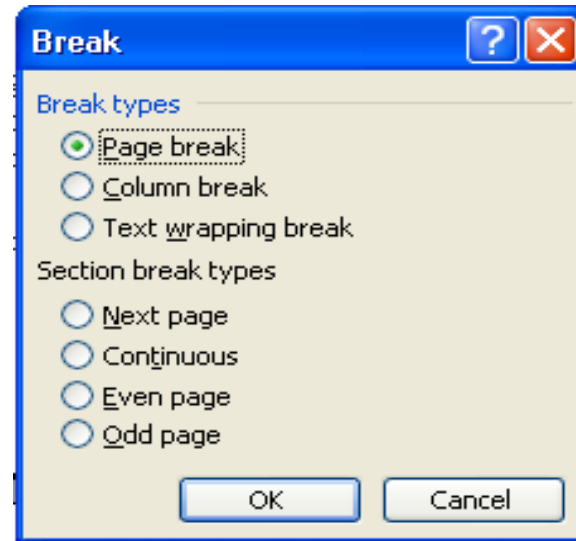
## **1.1 Proper Usage:**

- Should represent an independent function or application.
- Use to present constantly used window components and controls.
- Used frequently.
- Used for presenting information that is continually updated.
- Use for providing context for dependent windows to be created.

## **1.2 Guidelines:**

- Do not divide an independent function into two or more primary windows.
- Do not present unrelated functions in one primary window.
- Also referred to as the application window or the main window or the parent window.

## 2. Secondary Windows:



### 2.1 Proper Usage:

- For performing subordinate, supplemental, or ancillary actions that are:
  - ✓ Extended or more complex in nature.
  - ✓ Related to objects in the primary window.
- For presenting frequently or occasionally used window components.

## 2.2 Guidelines:

- Should typically not appear as an entry on the taskbar.
- A secondary window should not be larger than 263 dialog units x 263 dialog units.
- A ***dependent secondary window*** can only be displayed from a command on the interface of its primary window. Typically associated with a single data object, and appears on top of the active window when requested. It is movable, and scrollable.
- An ***independent secondary window*** can be opened independently of a primary window—for example, a property sheet displayed when the user clicks the Properties command on the menu of a desktop icon.

## **a. Modal and Modeless:**

### **Modal:**

- Use when interaction with any other window must not be permitted.
- Use for:
  - ✓ Presenting information (message box).
  - ✓ Receiving user input (prompt box).
  - ✓ Asking questions (question box).

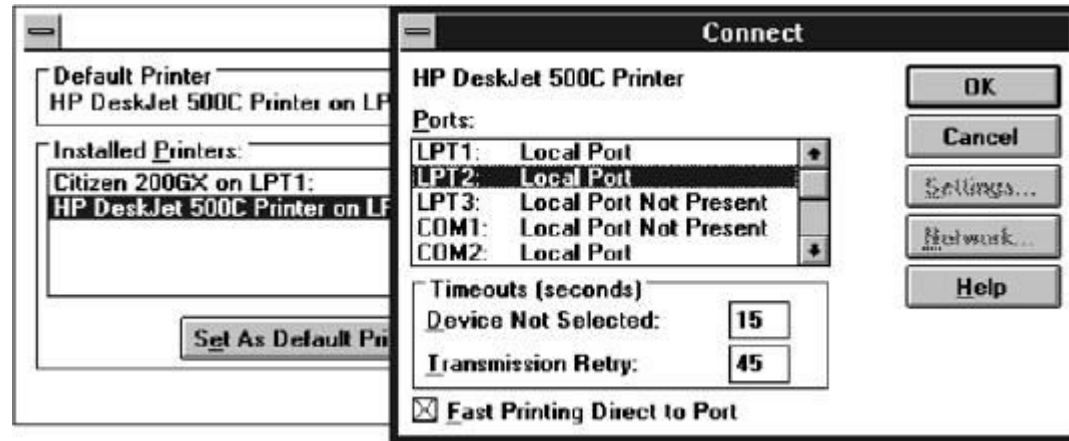
### **Modeless:**

- Use when interaction with other windows must be permitted.
- Use when interaction with other windows must be repeated.

## b.Cascading and Unfolding:

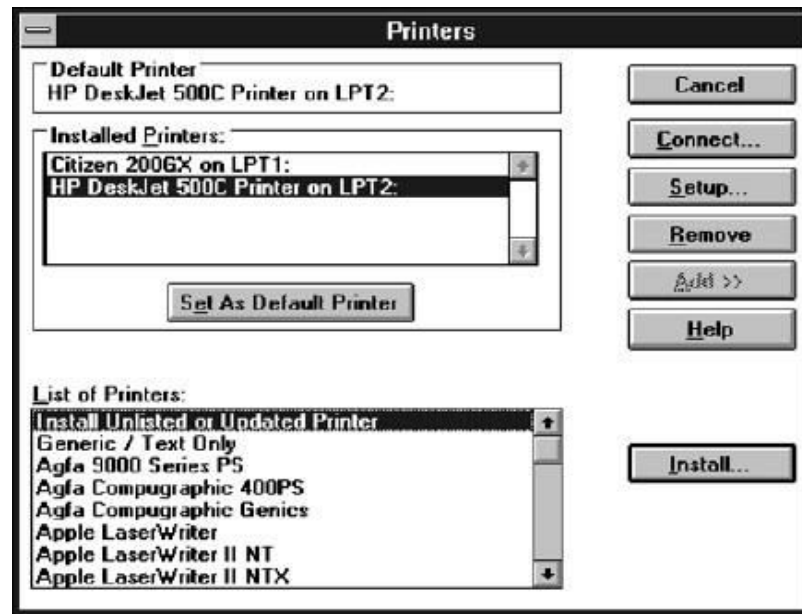
### Cascading:

- Purpose:
  - To provide advanced options at a lower level in a complex dialog.
- Guidelines:
  - Provide a “To a Window” indicator an ellipsis (. . . ).
  - Present the additional dialog box in cascaded form.
  - Provide no more than two cascades in a given path.



## Unfolding:

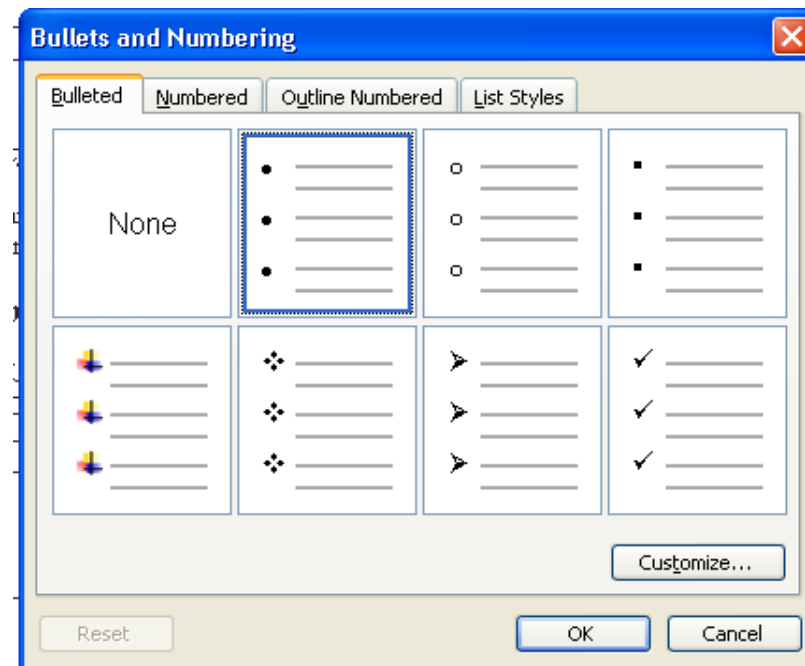
- Purpose:
  - To provide advanced options at the same level in a complex dialog.
- Guidelines:
  - Provide an expanding dialog symbol (>>).
  - Expand to right or downward.



### c. Property Sheets and Property Inspectors:

- Secondary windows provide two other techniques for displaying properties, *property sheets* and *property inspectors*.

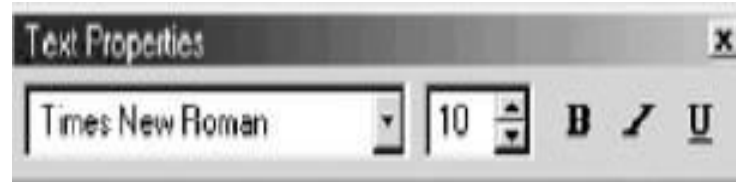
### Property Sheets:





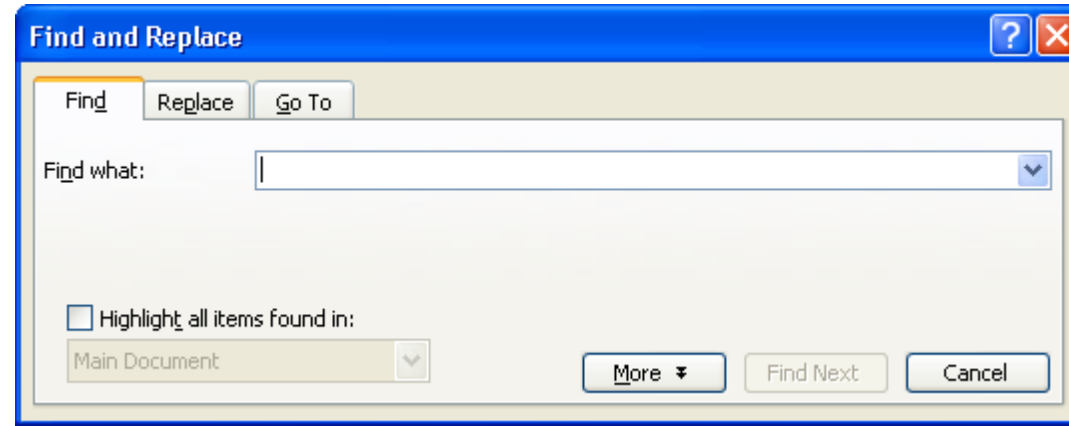
- Use for presenting the complete set of properties for an object.
- Categorize and group within property pages, as necessary.
- Command buttons to include:
  - ✓ OK.
  - ✓ Cancel.
  - ✓ Apply.
  - ✓ Reset.
  - ✓ Others as necessary.
- For single property sheets, place the commands on the sheet.
- For tabbed property pages, place the commands outside the tabbed pages.

## Property Inspectors:



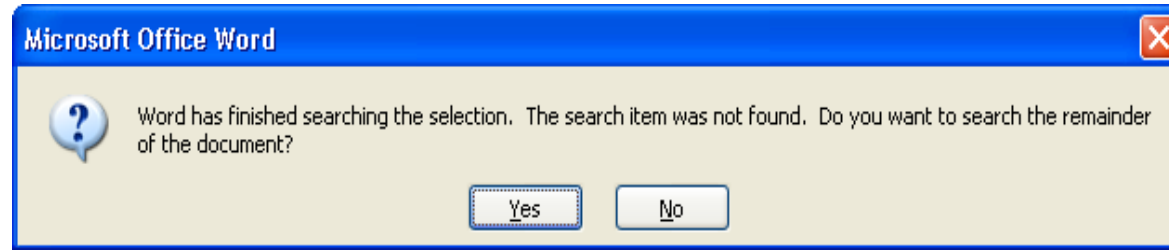
- Use for displaying only the most common or frequently accessed objects properties.
- Make changes dynamically.

## Dialog Boxes:



- Use for presenting brief messages.
- Use for requesting specific, transient actions.
- Use for performing actions that:
  - ✓ Take a short time to complete.
  - ✓ Are not frequently changed.
- Command buttons to include:
  - ✓ OK.
  - ✓ Cancel.
  - ✓ Others as necessary.

## Message Boxes:



- Use for displaying a message about a particular situation or condition.
- Command buttons to include:
  - ✓ OK.
  - ✓ Cancel.
  - ✓ Help.
  - ✓ Yes and No.
  - ✓ Stop.
  - ✓ Buttons to correct the action that caused the message box to be displayed.
- Enable the title bar close box only if the message includes a cancel button.

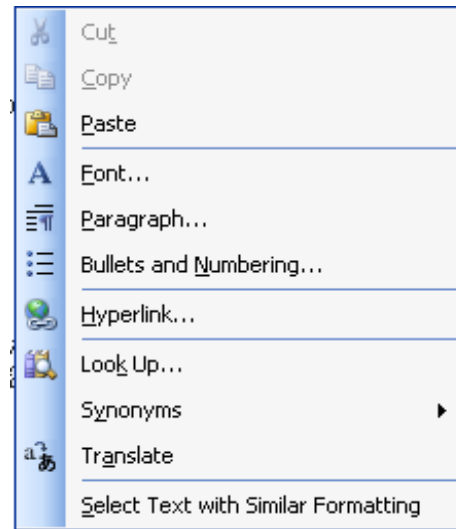
## Palette Windows:

- Use to present a set of controls.
- Design as resizable.



## Pop-up Windows:

- Use pop-up windows to display:
  - Additional information.
  - Textual labels for graphical controls.



# WINDOWS

UNIT III

# WINDOW MANAGEMENT:

- Microsoft Windows also provides several window management schemes,
  - ✓ Single document interface
  - ✓ Multiple-document interface
  - ✓ Workbooks
  - ✓ Projects



# 1. SINGLE-DOCUMENT INTERFACE:

- **Description:**

- A single primary window with a set of secondary windows.

- **Proper usage:**

- Where object and window have a simple, one-to-one relationship.
  - Where the object's primary presentation or use is as a single unit.
  - To support alternate views with a control that allows the view to be changed.
  - 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 :

- **Description:**

- A technique for managing a set of windows where documents are opened into windows.
- Contains:
  - A single primary window, called the parent.
  - A set of related document or child windows, each also essentially a primary window.
- Each child window is constrained to appear only within the parent window.

- **Proper usage:**

- To present multiple occurrences of an object.
- To compare data within two or more windows.

- To present multiple parts of an application.
- To clearly segregate the objects and their windows used in a task.

- **Advantages:**

- The child windows share the parent window's interface components (menus, toolbars, and status bars), making it a very space-efficient interface.
- Useful for managing a set of objects.

- **Disadvantages:**

- Creating confusion.
- The relationship between files and their windows is abstract, making an MDI application more challenging for beginning users to learn.
- Confining child windows to the parent window can be inconvenient or inappropriate for some tasks.
- The nested nature of child windows may make it difficult for the user to distinguish a child window in a parent window.

### **3. WORKBOOKS:**

- **Description:**

- A window or task management technique that consists of a set of views organized like a tabbed notebook.
- It is based upon the metaphor of a book or notebook.
- Views of objects are presented as sections within the workbook's primary windows; child windows do not exist.
- Each section represents a view of data.
- Tabs can be included and used to navigate between sections.
- Otherwise, its characteristics and behavior are similar to those of the multiple document interface with all child windows maximized.

- **Proper usage:**

- 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 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:

- **Description:**

- A technique that consists of a container: a project window holding a set of objects.
- The objects being held within the project window can be opened in primary windows that are peers with the project window.
- Visual containment of the peer windows within the project window is not necessary.
- Each opened peer window must possess its own menu bar and other interface elements.
- Each opened peer window can have its own entry on the task bar.
- 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.

- **Proper usage:**

- 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.
- 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.



# ORGANIZING WINDOW FUNCTIONS:

## WINDOW ORGANIZATION:

- Organize windows to support user tasks.
- Support the most common tasks in the most efficient sequence of steps.
- 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.

## **NUMBER OF WINDOWS:**

- Minimize the number of windows needed to accomplish an objective.
- The general rule:
  - Minimize the number of windows used to accomplish an objective.
  - Use a single window whenever possible.
  - Consider, however, 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.

# WINDOW OPERATIONS:

## 1. ACTIVE WINDOWS:

- A window should be made active with as few steps as possible.
- Visually differentiate the active window from other windows.

### Guidelines:

- Design easy to use and learn windowing operations.
- Minimize the number of window operations necessary to achieve a desired effect.
- Make navigating between windows particularly easy and efficient to do.
- Make the setting up of windows particularly easy to remember.

## 2. OPENING A WINDOW:

- Provide an iconic representation or textual list of available windows.
- When opening a window:
  - Position the opening window in the most forward plane of the screen.
  - Adapt the window to the size and shape of the monitor on which it will be presented.
  - Designate it as the active window.
  - Ensure that its title bar is visible.
- When a primary window is opened or restored, position it on top.
- 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.
- Display a window in the same state as when it was last accessed.
- With tiled windows, provide an easy way to resize and move newly opened windows.

### **3. SIZING WINDOW:**

- Provide large-enough windows to:
  - Present all relevant and expected information for the task.
  - Avoid hiding important information.
  - Avoid crowding or visual confusion.
  - Minimize the need for scrolling.
- Make the window as small as possible.
- Optimum window sizes:
  - For text, about 12 lines.
  - For alphanumeric information, about seven lines.

- **Advantages:**

- They permit displaying of important information.
- 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:**

- Longer pointer movements are required.
- Windows are more crowded.
- More visual scanning is required.
- It is not as easy to hide inappropriate data.

## 4. WINDOW PLACEMENT:

- In placing a window on the display, consider:
  - The use of the window.
  - The overall display dimensions.
  - The reason for the window's appearance.

### **Guidelines:**

- Position the window so it is entirely visible.
- If the window is being restored, place the window where it last appeared.
- If the window is new, and a location has not yet been established, place it:
  - At the point of the viewer's attention, usually the location of the pointer or cursor.
  - In a position convenient to navigate to.



- For multiple windows, a cascading presentation is recommended.
- Do not let the user move a window to a position where it cannot be easily repositioned.
- Horizontally center a secondary window within its primary window just below the title bar, menu bar, and any docked toolbars.
- Dialog boxes:
  - If the dialog box relates to the entire system, center it on screen.
  - Keep key information on the underlying screen visible.
  - If one dialog box calls another, make the new one movable whenever possible.

## 5. WINDOW SEPARATION:

- Crisply, clearly, and pleasingly demarcate a window from the background of the screen on which it appears.
  - Provide a surrounding solid line border for the window.
  - Provide a window background that sets the window off well against the overall screen background.
  - Consider incorporating a drop shadow beneath the window.

## **6. MOVING A WINDOW :**

- Permit the user to change the position of all windows.
- Change the pointer shape to indicate that the move selection is successful.
- Move the entire window as the pointer moves.
  - If it is impossible to move the entire window, move the window outline while leaving the window displayed in its original position.
- Permit the moving of a window without its being active.

## 7. RESIZING A WINDOW:

- Permit the user to change the size of primary windows.
  - Unless the information displayed in the window is fixed or cannot be scaled to provide more information.
- Change the pointer shape to indicate that the resizing selection is successful.
- The simplest operation is to anchor the upper-left corner and resize from the lower right corner.
  - Also permit resizing from any point on the window.
- Show the changing window as the pointer moves.
  - If it is impossible to show the entire window being resized, show the window's outline while leaving the window displayed in its original position.

- When window size changes and content remains the same:
  - Change image size proportionally as window size changes.
- If resizing creates a window or image too small for easy use, do one of the following:
  - Clip (truncate) information arranged in some logical structure or layout when minimum size is attained, or
  - When no layout considerations exist, format (restructure) information as size is reduced, or
  - Remove less useful information (if it can be determined), or
  - When minimum size is attained, replace information with a message that indicates that the minimum size has been reached and that the window must be enlarged to continue working.
- Permit resizing a window without its being active.

## **8. WINDOW SHUFFLING:**

- Window shuffling must be easy to accomplish.

## **9. KEYBOARD CONTROL/MOUSE LESS OPERATION:**

- Window actions should be capable of being performed through the keyboard as well as with a mouse.
- Keyboard alternatives should be designated through use of mnemonic codes as much as possible.
- Keyboard designations should be capable of being modified by the user.

## **10. CLOSING A WINDOW:**

- Close a window when:
  - The user requests that it be closed.
  - The user performs the action required in the window.
  - The window has no further relevance.
- If a primary window is closed, also close all of its secondary windows.
- When a window is closed, save its current state, including size and position, for use when the window is opened again.

# WINDOWS

UNIT III



# SCREEN BASED CONTROL:

- Screen-based controls, often simply called *controls* and sometimes called *widgets*, are the elements of a screen that constitute its body.
- By definition, they are graphic objects that represent the properties or operations of other objects.
- A control may:
  - Permit the entry or selection of a particular value.
  - Permit the changing or editing of a particular value.
  - Display only a particular piece of text, value, or graphic.
  - Cause a command to be performed.

- Three extremely important principles regarding controls should be noted:
  - A control must:
    - Look the way it works.
    - Work the way it looks.
  - A control must be used exactly as its design intended.
  - A control must be presented in a standard manner.
- The look of a control should make it obvious that it is a control.
- Its design characteristics should be “enterability” or “clickability.”

# OPERABLE CONTROL

- Operable controls are those that permit the entry, selection, changing, or editing of a particular value, or cause a command to be performed.
- Operable controls include
  - Buttons
  - Text entry/read-only control
  - Selection control
  - Combination control
  - Other specialized controls.

# 1. **BUTTONS:**

- **Description:**

- A square or rectangular-shaped control with a label inside that indicates action to be accomplished.
- The label may consist of text, graphics, or both.

- **Purpose:**

- To start actions.
- To change properties.
- To display a pop-up menu.

- **Advantages:**

- Always visible, reminding one of the choices available.
- Convenient.

- Larger size generally provides faster selection target.
- Can possess 3-D appearance.
- Can provide meaningful descriptions of the actions that will be performed.
- **Disadvantages:**
  - Consumes screen space.
  - Size limits the number that may be displayed.
  - Requires looking away from main working area to activate.
  - Requires moving the pointer to select.
- **Proper usage:**
  - Use for frequently used actions that are specific to a window.
    - To cause something to happen immediately.
    - To display another window.
    - To display a menu of options.

## 1.1 COMMAND BUTTON:

### Guidelines:

- **Usage:** Used to provide fast access to frequently used or critical commands.
- **Structure:** Provide a rectangular shape with the label inscribed within it.
- **Labels:** Label should be standard and provide meaningful descriptions of the actions that will be performed.
- Use single-word labels whenever possible.
- Do not number labels.
- **Size:** Provide as large a button as feasible.
- Maintain consistent button heights and widths.

- **Number:** Restrict the number of buttons on a window to six or fewer.
- **Location:** Maintain consistency in button location between windows.
- Never simply “fit” buttons in available space.
- Provide adequate spacing between buttons.
- **Organization:** Organize standard buttons in the manner recommended by the platform being used.
- Keep related buttons grouped together.
- Buttons for most frequent actions at the top and the left.
- **Keyboard Equivalents:** Assign a keyboard equivalent mnemonic to each button to facilitate keyboard selection.

- **Button Activation:**
- Pointing:
  - Highlight the button in some visually distinctive manner when the pointer is resting on it and the button is available for selection.
- Activation:
  - Call attention to the button in another visually distinctive manner when it has been activated or pressed.
  - If a button can be pressed continuously, permit the user to hold the mouse button down and repeat the action.



## 1.2 TOOL BAR:

- Toolbars are compilations of commands, actions, or functions, usually graphical in structure but sometimes textual, grouped together for speedy access.
- Toolbars may also be called *button bars*, *control bars*, or *access bars*.
- Specialized toolbars may also be referred to as *ribbons*, *toolboxes*, or *palettes*.

### Guidelines:

- **Usage:** To provide easy and fast access to most frequently used commands or options across multiple screens.

- **Structure:**
- **For Image:** Provide buttons of equal size. Create a meaningful and unique icon.
- **For Text:** Create a meaningful label.
- **Consistency:** Use the same icon throughout an application and between applications.
- **Organization:** Order the buttons based on common and customary grouping schemes.
- Keep related buttons grouped together.
- **Location:** Position main features and functions bar horizontally across top of window just below menu bar.

- **Customization:** Permit toolbars to be turned off by the user.
- **Keyboard Equivalents:** Assign keyboard equivalents to facilitate keyboard selection.
- **Button Activation:**
- Pointing:
  - Highlight the button in some visually distinctive manner when the pointer is resting on it and the button is available for selection.
- Activation:
  - Call attention to the button in another visually distinctive manner when it has been activated or pressed.

# TEXT ENTRY / READ ONLY CONTROL:

- A Text Entry/Read-Only control contains text that is exclusively entered or modified through the keyboard.
- It may also contain entered text being presented for reading or display purposes only.

# 1. TEXT BOXES:

- **Description:**

- A control, usually rectangular in shape, in which text may be entered or edited or displayed for read-only purposes.
- Usually possesses a caption describing the kind of information contained within it.
- An outline field border:
  - Is included for enterable/editable text boxes.
  - Is not included for read-only text boxes.
- Two types exist:
  - Single line.
  - Multiple line.
- When first displayed, the box may be blank or contain an initial value.

- **Purpose:**

- To permit the display, entering, or editing of textual information.
- To display read-only information.

- **Advantages:**

- Very flexible.
- Familiar.
- Consumes little screen space.

- **Disadvantages:**

- Requires use of typewriter keyboard.
- Requires user to remember what must be keyed.

- **Proper usage:**

- Most useful for data that is:
  - Difficult to categorize.
  - A variety of different lengths.
- When using a selection list is not possible.

## 1.1 TYPES OF TEXT BOX:

### ENTRY FILED:

- It consists of a rectangular box into which information is typed.
- It may also be referred to as an *edit* control.

### INQUIRY OR DISPLAY FIELD:

- Also rectangular in shape but contains text displayed purely for read-only purposes.

Entry/Modification:

Display/Read Only: Information

## 1.2 TWO FORMS OF TEXT BOX:

### SINGLE LINE TEXT BOX:

- **Description:**

- A control consisting of no more than one line of text.

- **Purpose:**

- To make textual entries when the information can be contained within one line of the screen

- **Typical uses:**

- Typing the name of a file to save.
- Typing the path of a file to copy.
- Typing variable data on a form.
- Typing a command.



## **MULTIPLE LINE TEXT BOX:**

- **Description:**

- A control consisting of a multiline rectangular box for multiple lines of text.

- **Purpose:**

- To type, edit, and read passages of text.

- **Typical uses:**

- Creating or reading an electronic mail message.
- Displaying and editing text files.

# SELECTION CONTROL:

- A selection control presents on the screen all the possible alternatives, conditions, or choices that may exist for an entity, property, or value.
- The relevant item or items are selected from those displayed.
- Selection controls include radio buttons, check boxes, list boxes, drop-down/pop-up list boxes, and palettes.

# 1. RADIO BUTTON:

- **Description:**

- A two-part control consisting of the following:
  - Small circles, diamonds, or rectangles.
  - Choice descriptions.
- When a choice is selected:
  - The option is highlighted.
  - Any existing choice is automatically unhighlighted and deselected.

- **Purpose:**

- To set one item from a small set of mutually exclusive options (2 to 8).

- **Advantages:**

- Easy-to-access choices.
- Easy-to-compare choices.
- Preferred by users.

- **Disadvantages:**

- Consume screen space.
- Limited number of choices.

- **Proper usage:**

- For setting attributes, properties, or values.
- For mutually exclusive choices (that is, only one can be selected).
- Where adequate screen space is available.
- Most useful for data and choices that are:
  - Discrete.
  - Small and fixed in number.

- **Guidelines:**
- **Choice Descriptions:** Provide meaningful and single line of text.
- **Size:** Show a minimum of two choices, a maximum of eight.
- **Structure:** A columnar orientation is the preferred manner of presentation. Left-align the buttons and choice descriptions.
- **Organization:** Arrange selections in expected order or follow other patterns such as frequency of occurrence, sequence of use, or importance.
- **Keyboard Equivalents:** Assign a keyboard mnemonic to each choice description.
- **Selection Method:**
  - **Pointing:** Highlight the selection choice in some visually distinctive way.
  - **Activation:** A radio button should be filled in with a solid dark dot.

## 2. CHECK BOX:

- **Description:**

- A two-part control consisting of a square box and choice description.
- Each option acts as a switch and can be either “on” or “off.”
  - When an option is selected (on), a mark such as an “X” or “check” appears within the square box, or the box is highlighted in some other manner.
  - Otherwise the square box is unselected or empty (off).
- Each box can be:
  - Switched on or off independently.
  - Used alone or grouped in sets.

- **Purpose:**

- To set one or more options as either on or off.

- **Advantages:**

- Easy-to-access choices.
- Easy-to-compare choices.
- Preferred by users.

- **Disadvantages:**

- Consume screen space.
- Limited number of choices.
- Single check boxes difficult to align with other screen controls.

- **Proper usage:**

- For setting attributes, properties, or values.
- For nonexclusive choices (that is, more than one can be selected).
- Where adequate screen space is available.
- Most useful for data and choices that are:
  - Discrete.
  - Small and fixed in number.

- **Guidelines:**
- **Choice Descriptions:** Provide meaningful and single line of text.
- **Size:** Show a minimum of one choices, a maximum of eight.
- **Structure:** A columnar orientation is the preferred manner of presentation. Left-align the buttons and choice descriptions.
- **Organization:** Arrange selections in logical order or follow other patterns such as frequency of occurrence, sequence of use, or importance.
- **Keyboard Equivalents:** Assign a keyboard mnemonic to each choice description.
- **Selection Method:**
  - **Pointing:** Highlight the selection choice in some visually distinctive way.
  - **Activation:** A check box should be filled in or made to look depressed or higher through use of a shadow.



### **3. PALETTES:**

- **Description:**

- A control consisting of a series of graphical alternatives. The choices themselves are descriptive, being composed of colors, patterns, or images.
- In addition to being a standard screen control, a palette may also be presented on a pull-down or pop-up menu or a toolbar.

- **Purpose:**

- To set one of a series of mutually exclusive options presented graphically or pictorially.

- **Advantages:**

- Pictures aid comprehension.
- Easy-to-compare choices.
- Usually consume less screen space than textual equivalents.

- **Disadvantages:**

- A limited number of choices can be displayed.
- Difficult to organize for scanning efficiency.
- Requires skill and time to design meaningful and attractive graphical representations.

- **Proper usage:**

- For setting attributes, properties, or values.
- For mutually exclusive choices (that is, only one can be selected).
- Where adequate screen space is available.
- Most useful for data and choices that are:
  - Discrete.
  - Frequently selected.
  - Limited in number.
  - Variable in number.

## Guidelines:

- **Graphical Representations:** Provide meaningful, accurate, and clear representations of choices. Create images of equal size.
- **Layout:** Create boxes large enough to:
  - Effectively illustrate the available alternatives.
  - Permit ease in pointing and selecting.
- **Organization:** Arrange palettes in expected or normal order.
- **Selection Method:**
  - **Pointing:** Highlight the choice in some visually distinctive way when the pointer or cursor is resting on it and the choice is available for selection.
  - **Activation:** When a choice is selected, distinguish it visually from the unselected choices by highlighting it in a different manner.

## 4. LIST BOXES:

- **Description:**

- A permanently displayed box-shaped control containing a list of attributes or objects from which:
  - A single selection is made (mutually exclusive), or
  - Multiple selections are made (non-mutually-exclusive).
- The choice may be text, pictorial representations, or graphics.
- Capable of being scrolled to view large lists of choices.

- **Purpose:**

- To display a collection of items containing:
  - Mutually exclusive options.
  - Non-mutually-exclusive options

- **Advantages:**

- Unlimited number of choices.
- Reminds users of available options.
- Box always visible.

- **Disadvantages:**

- Consumes screen space.
- Often requires an action (scrolling) to see all list choices.

- **Proper usage:**

- For selecting values or setting attributes.
- For choices that are:
  - Mutually exclusive (only one can be selected).
  - Non-mutually-exclusive (one or more may be selected).
- Where screen space is available.
- For data and choices that are:
  - Best represented textually.
  - Not frequently selected.
  - Large in number.

- **Guidelines:**
- **Selection Descriptions:** Clearly and meaningfully describe the choices available. Spell them out as fully as possible.
- **List Size:** Not actual limit in size.
- **Box Size:** Must be long enough to display 6 to 8 choices without requiring scrolling.
- **Layout:** Enclose the choices in a box with a solid border.
- **Captions:** Use mixed-case letters.
- **Selection Method:**
  - **Pointing:** Highlight the selection choice in some visually distinctive way when the pointer or cursor is resting on it and the choice is available for selection.
  - **Selection:** Use a reverse video or reverse color bar to surround the choice description when it is selected.

## 5. DROP DOWN/ POP UP LIST BOXES:

- **Description:**

- A single rectangular control that shows one item with a small button to the right side.

- **Purpose:**

- To select one item from a large list of mutually exclusive options when screen space is limited.

- **Advantages:**

- Unlimited number of choices.
- Reminds users of available options.
- Conserves screen space.

- **Disadvantages:**

- Requires an extra action to display the list of choices.
- When displayed, all choices may not always be visible, requiring scrolling.

- **Proper usage:**

- For selecting values or setting attributes.
- For choices that are mutually exclusive (only one can be selected).
- Where screen space is limited.
- For data and choices that are:
  - Best represented textually.
  - Infrequently selected.
  - Large in number.



- **Guidelines:**
- **Selection Descriptions:** Clearly and meaningfully describe the choices available. Spell them out as fully as possible.
- **List Size:** Not limited in size.
- **Box Size:** Long enough to display 6 to 8 choices without scrolling.
- **Organization:** Order in a logical and meaningful way to permit easy browsing.
- **Selection Method :**
  - **Pointing:** Highlight the selection choice in some visually distinctive way when the pointer or cursor is resting on it and the choice is available for selection.
  - **Activation:** Close the drop-down/pop-up list box when an item is selected.

# COMBINATION CONTROL:

- It is possible for a control to possess the characteristics of both a text field and a selection field.
- The types of combination entry/selection fields are
  - spin boxes
  - Combo boxes
  - drop-down/pop-up combination boxes.

# 1. SPIN BOXES:

- **Description:**

- A single-line field followed by two small, vertically arranged buttons.
  - The top button has an arrow pointing up.
  - The bottom button has an arrow pointing down.

- **Purpose:**

- To make a selection by either scrolling through a small set of meaningful predefined choices or typing text.

- **Advantages:**

- Consumes little screen space.
- Flexible, permitting selection or typed entry.

- **Disadvantages:**

- Difficult to compare choices.
- Can be awkward to operate.
- Useful only for certain kinds of data.

- **Proper usage:**

- For setting attributes, properties, or values.
- For mutually exclusive choices (only one can be selected).
- When the task requires the option of either key entry or selection from a list.
- Where screen space is limited.
- Most useful for data and choices that are:
  - Discrete.
  - Infrequently selected.
  - Small in number.

- **Guidelines:**
- **List Size:** Keep the list of items relatively short.
- **List Organization:** Order the list in the customary, consecutive, or expected order of the information contained within it.
- **Box size:** The spin box should be wide enough to display the longest entry or choice.
- **Caption:** Display it using mixed-case letters.
- **Entry and selection methods:**
  - Permit completion by:
    - Typing directly into the box.
    - Scrolling and selecting with a mouse.
    - Scrolling and selecting with the up/down arrow keys.

## 2. COMBO BOXES:

- **Description:**

- A single rectangular text box entry field, beneath which is a larger rectangular list box (resembling a drop-down list box) displaying a list of options.
- The text box permits a choice to be keyed within it.
- The larger box contains a list of mutually exclusive choices from which one may be selected for placement in the entry field.
- As text is typed into the text box, the list scrolls to the nearest match.

- **Purpose:**

- To allow either typed entry in a text box or selection from a list of options in a permanently displayed list box attached to the text box.

- **Advantages:**

- Unlimited number of entries and choices.
- Reminds users of available options.
- Flexible, permitting selection or typed entry.

- **Disadvantages:**

- Consumes some screen space.
- All list box choices not always visible, requiring scrolling.

- **Proper usage:**

- For entering or selecting objects or values or setting attributes.
- For information that is mutually exclusive (only one can be entered or selected).
- Where screen space is available.
- For data and choices that are:
  - Best represented textually.
  - Somewhat familiar or known.

- **Guidelines:**
- For the text box entry field, see “Text Box/Single Line” guidelines.
- For the list box, see “Drop-down/Pop-up List Box” guidelines.



### 3. DROP DOWN COMBO BOXES:

- **Description:**

- A single rectangular text box with a small button to the side and an associated hidden list of options.
- When requested, a larger associated rectangular box appears, containing a scrollable list of choices from which one is selected.
- Combines the capabilities of both a text box and a drop-down/pop-up list box.

- **Purpose:**

- To allow either typed entry or selection from a list of options in a list box that may be closed and retrieved as needed.

- **Advantages:**

- Unlimited number of entries and choices.
- Flexible, permitting selection or typed entry.
- Conserves screen space

- **Disadvantages:**

- Requires an extra step to display the list of choices.
- When displayed, all box choices may not always be visible, requiring scrolling.

- **Proper usage:**

- For entering or selecting objects or values or setting attributes.
- For information that is mutually exclusive (only one can be entered or selected).
- Where data must be entered that is not contained in the selection list.
- Where screen space is limited.
- For data and choices that are:
  - Best represented textually.
  - Somewhat familiar or known.
  - Large in number.

- **Guidelines:**
- For the text box entry field, see the “Text Box/Single Line” guidelines.
- For the box and selection components, see the “Drop-down/Pop-up List Box” guidelines.

# PRESENTATION CONTROL:

- Common presentation controls are
  - Static Text Fields
  - Group Boxes column headings
  - ToolTips
  - Balloon Tips
  - Progress Indicators
  - Scrolling Tickers
  - Sample Box

# 1. STATIC TEXT FIELD:

- **Description:**

- Read-only textual information.

- **Purpose:**

- To identify a control by displaying a control caption.
- To provide instructional or prompting information.
- To present descriptive information.

- **Proper usage:**

- To display a control caption.
- To display instructional or prompting information.
- To display descriptive information.

## **Guidelines:**

- **Captions:**

- Include a colon (:) as part of the caption.
- Include a mnemonic for keyboard access.
- When the associated control is disabled, display it dimmed.

- **Instructional or prompting information:**

- Display it in a unique and consistent font style for easy recognition and differentiation.

- **Descriptive information:**

- Follow all guidelines for required screen or control descriptive information.

## **2. GROUP BOXES:**

- **Description:**

- A rectangular frame that surrounds a control or group of controls.
- An optional caption may be included in the frame's upper-left corner.

- **Purpose:**

- To visually relate the elements of a control.
- To visually relate a group of related controls.

- **Proper usage:**

- To provide a border around radio button or check box controls.
- To provide a border around two or more functionally related controls.

- **Guidelines:**
- **Label or heading:**
  - Typically, use a noun or noun phrase for the label or heading.
  - Provide a brief label or heading, preferably one or two words.
  - Relate label or heading's content to the group box's content.
  - Capitalize the first letter of each significant word.
  - Do not include an ending colon ( : ).



### 3. TOOLTIPS:

- **Description:**

- A small pop-up window containing descriptive text that appears when a pointer is moved over a control or element either:
  - Not possessing a label.
  - In need of additional descriptive or status information.

- **Purpose:**

- To provide descriptive information about a control or screen element.

- **Advantages:**

- Identifies an otherwise unidentified control.
- Reduces possible screen clutter caused by control captions and descriptive information.
- Enables control size to be reduced.

- **Disadvantages:**

- Not obvious, must be discovered.
- Inadvertent appearance can be distracting.

- **Proper usage:**

- To identify a control that has no caption.
- To provide additional descriptive or status information about a screen element.

- **Guidelines :**

- Display after a short time-out.
- For toolbars, provide a brief word as a label.
- Present ToolTips at the lower-right edge of the pointer.
- Display them in the standard system ToolTip colors.
- Remove the ToolTip when the control is activated or the pointer is moved away.

## 4. BALLOON TIPS:

- **Description:**

- A small pop-up window that contains information in a word balloon.
- Components can include:
  - Title.
  - Body text.
  - Message Icons.
- Appear adjacent to the item to which they apply, generally above or to left.
- Only one tip, the last posted, is visible at any time.
- Tips are removed after a specified time period.

- **Purpose:**

- To provide additional descriptive or status information about a screen element.

- **Advantages:**

- Provides useful reminder and status information.

- **Disadvantages:**

- If overused they lose their attention-getting value.
  - If overused in situations the user considers not very important, their continual appearance can be aggravating.

- **Proper usage:**

- To display noncritical:
    - Reminder information.
    - Notification information.
  - Do not use tips to display critical information.

- **Guidelines:**

- Use a notification tip to inform the user about state changes.
- Use a reminder tip for state changes that the user might not usually notice.
- Point the tip of the balloon to the item it references.
- Do not use them to replace ToolTips.
- Do not overuse balloon tips.
- Restrict them to a length of 100 characters, including title and body text.

## 5. PROGRESS INDICATOR:

- **Description:**

- A rectangular bar that fills as a process is being performed, indicating the percentage of the process that has been completed.

- **Purpose:**

- To provide feedback concerning the completion of a lengthy operation.

- **Proper usage:**

- To provide an indication of the proportion of a process completed.

- **Guidelines:**
- When filling the indicator:
  - If horizontally arrayed, fill it from left to right.
  - If vertically arrayed, fill it from bottom to top.
- Fill it with a color or a shade of gray.
- Include descriptive text for the process, as necessary.
- Place text outside of the control.



## 6. SAMPLE BOX:

- **Description:**

- A box illustrating what will show up on the screen based upon the parameter or parameters selected.
- May include text, graphics, or both.

- **Purpose:**

- To provide a representation of actual screen content based upon the parameter or parameters selected.

- **Guidelines:**

- Include a brief label.
- Use mixed case in the headline style.
- Locate it adjacent to the controls upon which it is dependent.

## 7. SCROLLING TICKERS:

- **Description:**

- Text that scrolls horizontally through a container window.

- **Advantages:**

- Consume less screen space than full text.

- **Disadvantages:**

- Hard to read.
  - Time-consuming to interpret.
  - Distracting.

- **Guideline :**

- Do not use.

**CUSTOM CONTROL:**



# WINDOWS

UNIT III

# DEVICE BASED CONTROL

- Device-based controls often called input devices, are the mechanisms through which people communicate their desires to the system.

## **CHARACTERISTICS:**

- To point at an object on the screen.
- To select the object.
- To drag an object across the screen.
- To enter or manipulate data or information.

## **DIRECT DEVICES AND INDIRECT DEVICES:**

- Direct devices are operated on the screen itself.
- Examples:
  - Voice
  - Finger
  - Light Pen
- Indirect devices are operated in a location other than the screen.
- Examples:
  - Keyboard
  - Mouse
  - Track ball
  - Joystick

## **DIRECT DEVICES:**

### **VOICE:**

- **Description:**

- Automatic speech recognition by the computer.

- **Advantages:**

- Simple and direct.
- Useful for people who cannot use a keyboard.
- Useful when the user's hands are occupied.

- **Disadvantages:**

- High error rates due to difficulties in Recognizing spoken words.
- Slower throughput than with typing.
- Difficult to use in noisy environments.



## **FINGER:**

- **Description:**

- A special surface on the screen sensitive to finger or stylus touch.

- **Advantages:**

- Direct relationship between hand and pointer location in terms of direction, distance, and speed.
- Movement is direct.
- Requires no additional desk space.

- **Disadvantages:**

- Finger may hide the part of screen.
- Finger may be too large for accuracy with small objects.
- Very fatiguing to use for extended period of time.
- May damage the screen.

## **LIGHT PEN:**

- **Description:**

- A special surface on a screen sensitive to the touch of a special stylus or pen.

- **Advantages:**

- Direct relationship between hand and pointer movement in terms of direction, distance, and speed.
- Requires minimal additional desk space.
- More accurate than finger touching.

- **Disadvantages:**

- Hand may obscure part of screen.
- Requires picking it up to use.
- Very fatiguing to use for extended period of time.

# INDIRECT DEVICES:

## KEYBOARD:

- **Description:**

- Standard typewriter keyboard and cursor movement keys.

- **Advantages:**

- Familiar.
- Accurate.
- Use with very large screens.
- Useful for Entering text and alphanumeric data.

- **Disadvantages:**

- Slow for non-touch-typists.
- Slower than other devices in pointing.
- Requires discrete actions to operate.

# MOUSE:

- **Description:**

- A rectangular or dome-shaped, movable, desktop control containing from one to three buttons used to manipulate objects and information on the screen.
- Movement of screen pointer mimics the mouse movement.

- **Advantages:**

- Permits a comfortable hand resting position
- Does not obscure vision of the screen.

- **Disadvantages:**

- Requires hand to be removed from keyboard.
- Requires additional desk space.
- May require long movement distances.
- Requires a degree of eye-hand coordination.

## **TRACK BALL:**

- **Description:**

- A spherical object (ball) that rotates freely in all directions in its socket.
- Direction and speed is tracked and translated into cursor movement.

- **Advantages:**

- Does not obscure vision of screen.
- Does not require additional desk space (if mounted on keyboard).

- **Disadvantages:**

- Movement is indirect.
- Requires a degree of eye-hand coordination.
- Requires hand to be removed from keyboard keys.
- Requires different hand movements.
- May be difficult to control.
- May be fatiguing to use over extended time.

# JOYSTICK:

- **Description:**

- A stick or bat-shaped device anchored at the bottom.
- Variable in size, smaller ones being operated by fingers, larger ones requiring the whole hand.

- **Advantages:**

- Does not obscure vision of screen.
- Does not require additional desk space (if mounted on keyboard).

- **Disadvantages:**

- Movement is indirect.
- Requires a degree of eye-hand coordination.
- Requires hand to be removed from keyboard keys.
- Requires different hand movements.
- May be difficult to control.
- May be fatiguing to use over extended time.

## **SELECTING PROPER DEVICE BASED CONTROL:**

### **1. FACTOR FOR SELECTING:**

- Consider the characteristics of the task.
- Consider user characteristics and preferences.
- Consider the characteristics of the environment.
- Consider the characteristics of the hardware.
- Consider the characteristics of the device in relation to the application.
- Provide flexibility.
- Minimize eye and hand movements between devices.

## 2. POINTER GUIDELINES:

- The pointer:
  - Should be visible at all times.
  - Should contrast well with its background.
  - Should maintain its size across all screen locations and during movement.
- The user should always position the pointer.