Scheme of Evaluation

VTU Semester End Exam(SEE) – January 2025 – Answer Key

Sub:	User Interface Design					**Code:**		21IS733
Date**:**	15/01/2025	Duration:	180 mins	Max Marks:	100	**Sem:**	VII	**Branch:** ISE

Note: Answer Any five full questions, choosing one full question from each module.

Question #		**Description**	**Max Marks**	
		MODULE-1		
1	a) b) c)	What is user interface design? Elaborate its importance and benefits. Discuss about the concept of direct manipulation. Provide any 10 differences between GUI and web design. **(OR)**	10M 5M 5M	**20M**
2	a) b) c)	Describe the characteristics of Graphical User Interface, Bring out the differences between internet and intranet. Explain the principles of XEROX STAR.	10M 5M 5M	**20M**
		MODULE-2		
3	a) b) c)	Discuss about obstacles and pitfalls in design along with five commandments. Explain the common usability problems. Discuss the method for gaining understanding of users. **(OR)**	10M 5M 5M	**20M**
4	a) b) c)	Describe the techniques for determining user requirements using direct methods. Provide the guidelines for designing the conceptual model. Discuss about the usage of objects and metaphors in design.	10M 5M 5M	**20M**
		MODULE-3		

5	a) b) c)	With diagrams, illustrate the structure of menus. Describe different functions performed by the menus. Discuss about the content of menus. **(OR)**	10M 5M 5M	**20M**
6	a) b) c)	List all graphical menus and explain any one in detail Illustrate the concept of formatting the menus Describe any 3 guidelines in the phrasing of menus	10M 5M 5M	**20M**
		MODULE-4		
7	a) b) c)	With diagrams, illustrate the different components of a window. Provide various advantages of using windows Explain the characteristics of windows **(OR)**	10M 5M 5M	**20M**
8	a) b) c)	Discuss about different presentation styles of a window. Provide guidelines for selecting the device-based controls. Explain the characteristics of touch screen and keyboard.	10M 5M 5M	**20M**
		MODULE-5		
9	a) b) c)	What are operable controls? Elaborate the usage of command buttons with guidelines. Discuss about tool tips and progress indicator controls. Briefly explain Radio Button control. **(OR)**	10M 5M 5M	**20M**
10	a) b) c)	What is the purpose of using prototypes? Explain any two types of prototypes. Discuss about the concept of Cognitive walkthrus. Write short notes on Usability Testing.	10M 5M 5M	**20M**

Q. 1 a) What is user interface design? Elaborate its importance and benefits.

User Interface Design:

- User interface design is a subset of a field of study called human-computer interaction (HCI). Human-computer interaction is the study, planning, and design of how people and computers work together so that a person's needs are satisfied in the most effective way. The user interface is the part of a computer and its software that people can see, hear, touch, talk to, or otherwise understand or direct. User interface has 2 components: Input, Output. Input is how a person communicates his or her needs or desires to the computer. Ex. Keybord,mouse. Output is how the computer conveys its results of its computations and requirements to the user. Ex. Display screen.

Importance of good design:

With today's technology and tools, and our motivation to create really effective and usable interfaces and screens, why do we continue to produce systems that are inefficient and confusing or, at worst, just plain unusable? Is it because: 1. We don't care? 2. We don't possess common

sense? 3. We don't have the time? 4. We still don't know what really makes good design? A well-designed interface and screen is terribly important to our users. It is their window to view the capabilities of the system. To many, it is the system, being one of the few visible components of the product we developers create. It is also the vehicle through which many critical tasks are presented. These tasks often have a direct impact on an organization's relations with its customers, and its profitability.

Benefits of good design:

Training costs are lowered because training time is reduced. • Support line costs are lowered because fewer assist calls are necessary. • Employee satisfaction is increased because aggravation and frustration are reduced. • Customers benefit because of the improved service they receive. Based on an actual system requiring processing of 4.8 million screens per year and illustrated in Table 1.1

Table 1.1 Impact of Inefficient Screen Design on Processing Time

ADDITIONAL SECONDS REQUIRED PER SCREEN IN SECONDS	ADDITIONAL PERSON-YEARS REQUIRED TO PROCESS 4.8 MILLION SCREENS PER YEAR
1	.7
5	3.6
10	7.1
20	14.2

Studies have also shown that the proper formatting of information on screens does have a significant positive effect on performance. The benefits of a well-designed screen have also been under experimental scrutiny.
A general rule of thumb: every dollar invested in usability returns $10 to $100 (IBM, 2001).

Q. 1 b) Discuss about the concept of direct manipulation.

The Concept of Direct Manipulation

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 already familiar with the objects and actions in his or her environment of interest. The system simply replicates them and portrays them on a different medium, the screen. A person has the power to access and modify these objects, among which are windows. A person is allowed to work in a familiar environment and in a familiar way, focusing on the data, not the application and tools.
Continuous visibility of objects and actions : Like one's desktop, objects are continuously visible. Reminders of actions to be performed are also obvious, labeled buttons replacing complex syntax and command names. Cursor action and motion occurs in physically obvious and intuitively natural ways.

Actions are rapid and incremental with visible display of results : Since tactile feedback is not yet possible (as would occur with one's hand when one touches something), 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, and 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.

Q. 1 c) Provide any 10 differences between GUI and web design.

- A Web interface possesses a number of characteristics, some similar to a GUI interface, and, as has already been shown, some different.

GUI versus Web Design

Characteristics	GUI	WEB
Devices	User hardware variations limited. User hardware characteristics well defined Screens appear exactly as specified.	User hardware variations enormous. Screen appearance influenced by hardware being used.
User Focus	Data and applications.	Information and navigation.
Data	Typically created and used by known and trusted	Full of unknown content.
Information	Sources are trusted. Properties generally known. Typically placed into system by users or known people and organizations.	Source not always trusted. Often not placed onto the Web by users or known people and organizations.
User Tasks	Install, configure, personalize, start, use, and Open, use, and close data files. Familiarity with applications often achieved.	Link to a site, browse or read pages, fill out forms, upgrade programs. register for services, participate in transactions, download and save things.
Presentation	Windows, menus, controls, data, toolbars Presented as specified by designer. Generally standardized by toolkits and style specifications. guides.	Two components, browser and page. Within page, any combination of text, images, audio, video, and animation. May not be presented as specified by the designer—dependent on browser, monitor, and user Little standardization.
Navigation	Through menus, lists, trees, dialogs, and wizards.	Through links, bookmarks, and typed URLs.
Interaction	Interactions such as clicking menu choices, pressing buttons, selecting list choices, and cut/copy/paste occur within context of active program.	Basic interaction is a single click. This can cause extreme changes in context, which may not be noticed.
Response Time	Nearly instantaneous	Quite variable, depending on transmission speeds, page content, and so on. Long times can upset the user.
System Capability	Unlimited capability proportional to sophistication of hardware and software.	Limited by constraints imposed by the hardware, browser, software, client support, and user willingness to allow features because of response time, security, and privacy concerns.
Task Efficiency	Targeted to a specific audience with specific tasks. Only limited by the amount of programming undertaken to support it.	Limited by browser and network capabilities. Actual user audience usually not well understood. Often intended for anyone and everyone.

Characteristics	GUI	WEB
Consistency	Major objective exists within and across applications.Aided by platform toolkit and design guidelines. Universal consistency in GUI products generally	Sites tend to establish their own identity.Frequently standards set within a site. Frequent ignoring of GUI guidelines for identical created through toolkits and design guidelines. components, especially controls.
User Assistance	Integral part of most systems and applications. Documentation, both online and offline, Customer service support, if provided, usually provided. Personal support	No similar help systems. Accessed through standard mechanisms. The little available help is built into the page oriented to product or service offered.
Integration	dSeamlesssk also usuallyintegration provided. of all applications into the platform environment is a major objective.	Apparent for some basic functions within most Web sites (navigation, printing, and so on.) in accomplishing this objective. Sites tend to achieve individual distinction rather than integration.
Security	Tightly controlled, proportional to degree of willingness to invest resources and effort. Not an issue for most home PC users.	Renowned for security exposures. Browser-provided security options typically understood by average users. When employed, may have functionlimiting side effects.
Reliability	Tightly controlled in business systems,	Susceptible to disruptions caused by user, telephone proportional to degree of willingness line and cable providers, Internet service providers, to invest resources and effort. hosting servers, and remotely accessed sites.

Q.2 a) Describe the characteristics of Graphical User Interface.

1. Sophisticated Visual Presentation
2. Pick-and-Click Interaction
3. Restricted Set of Interface Options
4. Visualization
5. Object Orientation
6. Use of Recognition Memory
7. Concurrent Performance of Functions

1. 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 displaying lines, including drawings and icons. It also permits the displaying of a variety of character fonts, including different sizes and styles. The display of 16 million or

more colors is possible on some screens. Graphics also permit animation and the presentation of 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 -
- represent objects such as programs or files Assorted screen-based controls -
- text boxes, list boxes, combination boxes, settings, scroll bars, and buttons
- 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.

2. Pick-and-click interaction:

To identify the element for 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. The user moves the mouse pointer to the relevant element (pick) and the action is signaled (click).

The secondary mechanism for performing these selection actions is the keyboard. Most systems permit pick-and-click to be performed using the keyboard as well.

3. A 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, nothing more. This concept fostered the acronym WYSIWYG.

4. visualization:

Visualization is a cognitive process that allows people to understand information that is difficult to perceive, because it is **either too voluminous or too abstract**. It involves changing an entity's representation to reveal gradually the structure and/or function of the underlying system or process. Presenting specialized graphic portrayals facilitates visualization.

The best visualization method for an activity depends on what people are trying to learn from the data. The goal is not necessarily to reproduce a realistic graphical image, but to produce one that conveys the most relevant information. **Effective visualizations can facilitate mental insights, increase productivity and more accurate use of data.**

5. Object orientation:

A graphical system consists of **objects and actions**. *Objects* are what people see on the screen.Objects can be composed of *subobjects.*

A) IBM's Common User Access application breaks objects into three meaningful classes:

1.Data
2.Container
3.Device.

Data objects present information. This information, either text or graphics, normally appears in the body of the screen. It is, essentially, the screen-based controls for information collection or presentation organized on the screen.
Container objects are objects to hold other objects. They are used to group two or more related objects for easy access and retrieval. There are three kinds of container objects:

Workplace Folders / Workareas.

The *workplace* is the desktop, the storage area for all objects. *Folders* are general-purpose containers for long-term storage of objects. *Workareas* are temporary storage folders used for storing multiple objects currently being worked on.
Device objects represent physical objects in the real world, such as printers or trash baskets. These objects may contain others for acting upon. A file, for example, may be placed in a printer for printing of its contents.
B) Microsoft Windows specifies the characteristics of objects depending upon the relationships that exist between them.These relationships are called collections, constraints, composites, and containers.
A ***collection*** is the simplest relationship—the objects sharing a common aspect. A collection might be the result of a query or a multiple selection of objects. Operations can be applied to a collection of objects.
A ***constraint*** is a stronger object relationship. Changing an object in a set affects some other object in the set. A document being organized into pages is an example of a constraint. A ***composite*** exists when the relationship between objects becomes so significant that the aggregation itself can be identified as an object. Examples include a range of cells organized into a spreadsheet, or a collection of words organized into a paragraph. A ***container*** is an object in which other objects exist. Examples include text in a document or documents in a folder. A container often influences the behavior of its content. It may add or suppress certain properties or operations of objects placed within it, control access to its content, or control access to kinds of objects it will accept.
Another important object characteristic is ***persistence***. Persistence is the maintenance of a state once it is established. An object's state (for example, window size, cursor location, scroll position, and so on) should always be automatically preserved when the user changes it. **Properties or Attributes of Objects**
Properties help to describe an object and can be changed by users. Examples of properties are text styles (such as normal or italics), font sizes (such as 10 or 12 points), or window background colors (such as black or blue). **Actions**
Commands are actions that manipulate objects. They may be performed in a variety of ways, including by direct manipulation or through a command button. They are executed immediately when selected. Once executed, they cease to be relevant. Examples of commands are opening a document, printing a document, closing a window, and quitting an application. *Property/attribute specification* actions establish or modify the attributes or properties of objects. When selected, they remain in effect until deselected. Examples include selecting cascaded windows to be displayed, a particular font style, or a particular color.
The following is a typical *property/attribute specification sequence:*

1. The user selects an object—for example, several words of text.
2. The user then selects an action to apply to that object, such as the action BOLD.
3. The selected words are made bold and will remain bold until selected and changed again. A series of actions may be performed on a selected object. Performing a series of actions on an object also permits and encourages system

learning through exploration. **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.
Composed views present information and the objects contained within an object. They are typically associated with data objects and are specific to tasks and products being worked with.
Contents views list the components of objects. ***Settings*** views permit seeing and changing object properties. ***Help*** views provide all the help functions.

6. Use of recognition memory:

Continuous visibility of objects and actions encourages use of a person's more powerful recognition memory. The "out of sight, out of mind" problem is eliminated.

7. Concurrent performance of functions:

Graphic systems may do two or more things at one time. Multiple programs may run simultaneously. When a system is not busy on a primary task, it may process background tasks (cooperative multitasking). When applications are running as truly separate tasks, the system may divide the processing power into time slices and allocate portions to each application (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.

Q. 2 b) Bring out the differences between internet and intranet.

Characteristics of an Intranet versus the Internet

An intranet has many of the same characteristics as the Internet. They differ, however, in some important ways. The following discussion is partly based upon Nielsen (1997b):

Users. The users of intranets, being organization employees, know a lot about the organization, its structure, its products, its jargon, and its culture. Internet sites are used by customers and others who know much less about the organization, and often care less about it. The intranet user's characteristics and needs can be much more specifically defined than those of the general Internet user.

Tasks. An intranet is used for an organization's everyday activities, including complex transactions, queries, and communications. The Internet is mainly used to find information, with a supplementary use being simple transactions.

Type of information. An intranet contains detailed information needed for organizational functioning. Information is often be added or modified. The Internet usually presents more stable information: marketing and customer or client information, reports, and so forth.

Amount of information. Typically, an intranet site is much larger than an organization's Internet site. Massive amounts of information and processes seem to be needed to make an organization function. It has been estimated that an intranet site can be ten to one hundred times larger than its corresponding public site.

Hardware and software. Because intranets exist in a controlled environment, the kinds of computers, monitors, browsers, and other software can be restricted or standardized. The need for cross-platform compatibility is minimized or eliminated, permitting more predictable design. Upgraded communications also permit intranets to run from a hundred to a thousand times faster than typical Internet access can. This allows the use of rich graphics and multimedia, screen elements that contribute to very slow download times for most Internet users.

Design philosophy. Implementation on the intranet of current text-based and GUI applications will present a user model similar to those that have existed in other domains. This will cause a swing back to more traditional GUI designs — designs that will also incorporate the visual appeal of the Web, but eliminate many of its useless, promotional, and distracting features. The resulting GUI hybrids will be richer and much more effective.

Some specific intranet design guidelines are discussed in Part 2 of this book.

Q 2. c) Explain the principles of XEROX STAR

The Xerox STAR system was designed using a set of principles that became the foundation for graphical interfaces. These principles are:

- The illusion of manipulable objects:
    - The system creates displayable objects that can be selected and manipulated.
    - These objects should be represented in a way that is meaningful and appropriate for the application.
    - It should be obvious that the objects can be selected, and how to select them.
    - It should be clear that the selected object will be the focus of the next action.
    - This concept was described as "graphics with handles on it".
    - Stand-alone icons fulfill this requirement, with window handles in the borders for window-specific commands, pop-up menus and scroll bars.

- Visual order and viewer focus:
    - Attention must be drawn to the important and relevant elements of the display at the proper time.
    - Effective visual contrast is used to achieve this goal. Note that the STAR system was monochromatic, so colour was not used.
    - Animation and sound can also be used to draw attention.
    - Feedback should be provided to the user.
    -
    - \The pointer is usually the focus of the viewer's attention and is a useful mechanism for providing feedback (by changing shapes).

These principles were developed over the system's lengthy development process. They established the foundation for graphical interfaces.

Q.3 a) Discuss about obstacles and pitfalls in design along with five commandments.

Obstacles and Pitfalls in the Development Path - The path is littered with obstacles and traps, many of them human in nature. Gould (1988) has made these general observations about design:

- Nobody ever gets it right the first time.
- Development is chock-full of surprises.
- Good design requires living in a sea of changes.
- Making contracts to ignore change will never eliminate the need for change.
- Even if you have made the best system humanly possible, people will still make mistakes when using it.
- Designers need good tools.
- You must have behavioral design goals like performance design goals.

Common pitfalls are:

- No early analysis and understanding of the user's needs and expectations.
- A focus on using design features or components that are "neat" or "glitzy."
- Little or no creation of design element prototypes.
- No usability testing.
- No common design team vision of user interface design goals.
- Poor communication between members of the development team

Designing for People: The Five Commandments

Gain a complete understanding of users and their tasks.

The users are the customers. Today, people expect a level of design sophistication from all interfaces, including Web sites. The product, system or Web site must be geared to people's needs, not those of the developers. A wide gap in technical abilities, goals, and attitudes often exists between users and developers. A failure to understand the differences will doom a product or system to failure.

Solicit early and ongoing user involvement.

Involving the users in design from the beginning provides a direct conduit to the knowledge they possess about jobs, tasks, and needs. Involvement also allows the developer to confront a person's resistance to change, a common human trait. People dislike change for a variety of reasons, among them fear of the unknown and lack of identification with the sys-tem. Involvement in design removes the unknown and gives the user a stake in the system or identification with it. One caution, however: user involvement should be based on job or task knowledge, not status or position. The boss seldom knows what is really happening out in the office.

Perform rapid prototyping and testing.

Prototyping and testing the product will quickly identify problems and allow you to develop solutions. The design process is complex and human behavior is still not well understood. While the design guidelines that follow go a long way toward achieving ease of use, all problems cannot possibly be predicted. Prototyping and testing must be continually performed during all stages of development to uncover all potential defects. If thorough testing is not performed before product release, the testing will occur in the user's office.

Encountering a series of problems early in system use will create a negative first impression in the customer's mind, and this may harden quickly, creating attitudes that may be difficult to change. It is also much harder and more costly to fix a product after its release. In many instances, people may adapt to, or become dependent upon, a design, even if it is inefficient. This also makes future modifications much more difficult.

Modify and iterate the design as much as necessary. While design will proceed through a series of stages, problems detected in one stage may force the developer to revisit a previous stage. This is normal and should be expected. Establish user performance and acceptance criteria and continue testing and modifying until all design goals are met.

Integrate the design of *all* the system components. The software, the documentation, the help function, and training needs are all important elements of a graphical system or Web site and all should be developed concurrently. A system is being constructed, not simply software. Concurrent development of all pieces will point out possible problems earlier in the design process, allowing them to be more effectively addressed. Time will also exist for design trade-offs to be thought out more carefully.

Q.3 b) Explain the common usability problems.

Several factors contribute to usability problems in user interface design, including issues with user understanding, design consistency, and error management. These problems can manifest in various ways, impacting user experience and efficiency....

One key area of concern is **how easily users can understand and interact with the interface**....

Ambiguous menus and icons can hinder navigation. Design elements that lack clear functionality or use developer terminology instead of user-friendly language can also cause confusion. A system should be understandable, flowing in a comprehensible, obvious, and meaningful order. Strong clues to the operation of objects should be presented.

Inconsistency in design is another significant source of usability problems. This can include visual inconsistency in screen detail presentation, lack of restraint in design features, and overuse of three-dimensional presentations or bright colours. Design inconsistency has not disappeared with the Web; it has been magnified. The business

system user may visit a handful of systems in one day, the Web user may visit dozens, or many more.

Problems also arise from **how a system handles errors and provides feedback**. Inadequate feedback and confirmation, as well as a lack of system anticipation and intelligence, can lead to user frustration1. Error messages that are generic or poorly written can also frustrate users. A well-designed system should tolerate common human errors, prevent catastrophic errors, and provide constructive messages when errors occur.

Usability problems often stem from **a flawed design process**, including a failure to address critical design issues, an improper focus of attention, or development team organization failures. Common pitfalls include:

- No early analysis and understanding of the user's needs and expectations.
- A focus on using design features or components that are "neat" or "glitzy".
- Little or no creation of design element prototypes.
- No usability testing.
- No common design team vision of user interface design goals.
- Poor communication between members of the development team.

Additional usability problems include:

- Languages that permit only single-direction movement through a system.
- Input and direct manipulation limits.
- Highlighting and selection limitations.
- Unclear step sequences.
- More steps to manage the interface than to perform tasks.
- Complex linkage between and within applications.
- Inadequate error messages, help, tutorials, and documentation.

Usability can be assessed through observation and user feedback, watching for issues such as frequent questions or manual referrals. Practical measures of usability include time to complete a task, percentage of task completed, and the ratio of successes to failures.

Q.3 c) Discuss the method for gaining understanding of users.

To gain a thorough understanding of users, several direct and indirect methods can be employed. Gould (1988) suggests techniques to understand users, their tasks and needs, the organizations where they work, and the environments where the system will be used.

The objective of requirement analysis is to establish the need for a system. A requirement is an objective that must be met. A product description is developed and refined, based on input from users or marketing. There are many techniques for capturing information for determining requirements. Some of the techniques are listed below:

Individual Face-to-Face Interview

- A one-on-one visit with the user to obtain information. It may be structured or somewhat open-ended.
- A formal questionnaire should not be used, however. Useful topics to ask the user to describe in an interview include:
  - The activities performed in completing a task or achieving a goal or objective.
  - The methods used to perform an activity.
  - What interactions exist with other people or systems?
  - It is also very useful to also uncover any:
    - Potential measures of system usability

o Unmentioned exceptions to standard policies or procedures.

o Relevant knowledge the user must possess to perform the activity.

Telephone Interview or Survey

• A structured interview conducted via telephone.

• Advantages

• Arranging the interview in advance allows the user to prepare for it.

• Telephone interviews are less expensive and less invasive than personal interviews.

• They can be used much more frequently and are extremely effective for very specific information.

• Disadvantages

o It is impossible to gather contextual information, such as a description of the working environment, replies may be easily influenced by the interviewer's comments, and body language cues are missing.

o Also, it may be difficult to contact the right person for the telephone interview.

Traditional Focus Group

• A small group of users and a moderator brought together to verbally discuss the requirements.

• The purpose of a focus group is to probe user's experiences, attitudes, beliefs, and desires, and to obtain their reactions to ideas or prototypes

• Setting up focus group involves the following:

o Establish the objectives of the session.

o Select participants representing typical users, or potential users.

o Write a script for the moderator to follow.

o Find a skilled moderator to facilitate discussion, to ensure that the mdiscussion remains focused on relevant topics, and to ensure that everyone participates.

o Allow the moderator flexibility in using the script.

o Take good notes, using the session recording for backup and clarification

Facilitated Team Workshop

• A facilitated, structured workshop held with users to obtain requirements information similar to the traditional Focus Group

• Like focus groups, they do require a great deal of time to organize and run.

Observational Field Study

• Users are observed and monitored for an extended time to learn what they do.

• Observation provides good insight into tasks being performed, the working environment and conditions, the social environment, and working practices

• Observation, however, can be time-consuming and expensive.

• Video recording of the observation sessions will permit detailed task analysis.

Requirements Prototyping

• A demo, or very early prototype, is presented to users for comments

concerning functionality.

User-Interface Prototyping

A demo, or early prototype, is presented to users to uncover user-interface issues and problems

Usability Laboratory Testing

Users at work are observed, evaluated, and measured in a specially constructed laboratory to establish the usability of the product at that point in time.

Usability tests uncover what people actually do, not what they think they do a common problem with verbal descriptions

The same scenarios can be presented to multiple users, providing comparative data from several users.

INDIRECT METHODS
MIS Intermediary • A company representative defines the user's goals and needs to designers and developers.
Paper Survey or Questionnaire • A survey or questionnaire is administered to a sample of users using traditional mail methods to obtain their needs.
Electronic Survey or Questionnaire • A survey or questionnaire is administered to a sample of users using e-mail or the Web to obtain their needs.
Electronic Focus Group • A small group of users and a moderator discuss the requirements online using workstations.
Marketing and Sales • Company representatives who regularly meet customers obtain suggestions or needs, current and potential.
Support Line • Information collected by the unit that helps customers with day-to-day problems is analyzed (Customer Support, Technical Support, Help Desk, etc.).
E-Mail or Bulletin Board • Problems, questions, and suggestions from users posted to a bulletin board or through e-mail are analyzed.
User Group • Improvements are suggested by customer groups who convene periodically to discuss software usage.
Competitor Analyses • A review of competitor's products or Web sites is used to gather ideas, uncover design requirements and identify tasks.
Trade Show • Customers at a trade show are presented a mock-up or prototype and asked for comments.
Other Media Analysis • An analysis of how other media, print or broadcast, present the process, information, or subject matter of interest.
System Testing • New requirements and feedback are obtained from ongoing product testing

Q.4 a) Describe the techniques for determining user requirements using direct methods.

Refer Direct Methods in the above answer for Q.3c

4 b) Provide the guidelines for designing the conceptual model.

When designing conceptual models, several guidelines can aid in the development of a user-friendly system. These guidelines help facilitate the development of a useful mental model for the user.

Key guidelines for designing conceptual models:

- **Reflect the user's mental model, not the designer's** A user's expectations and knowledge differ from the designer's. The user concentrates on task performance and business objectives, while the designer focuses on interface design, objects, interaction methods, and screen representations.
- **Draw physical analogies or present metaphors**. A metaphor is a concept where knowledge about one thing is used to understand something else, acting as building blocks for understanding a system's organisation. Choose analogies that work best for objects and actions, using real-world metaphors. Replicate familiar actions but improve them if possible. Use simple and common metaphors, but test them to ensure they match expectations and are easily understood.
- **Comply with expectancies, habits, routines, and stereotypes** Create a system that builds on existing knowledge and habits. Use familiar associations and avoid unfamiliar. Use colours, words, and symbols in customary ways.
- **Provide action-response compatibility**. System responses should align with the actions that elicit them. Command names should reflect the actions that occur. Key organisation in documentation should reflect the keyboard layout.
- **Make invisible parts and processes of a system visible**. Reveal hidden system parts and processes to aid understanding. New system users often make incorrect assumptions about invisible elements, leading to faulty mental models.
- **Provide proper and correct feedback**.
- **Avoid anything unnecessary or irrelevant**. Do not display irrelevant information as it can lead to false mental models. This includes unneeded data fields, screen controls, or system status codes. Avoid overusing display techniques.
- **Provide design consistency**. Consistency reduces the number of concepts to learn, while inconsistency requires mastering multiple models. Explain any unavoidable inconsistencies to prevent users from assuming their model is incorrect.
- **Provide documentation and a help system that will reinforce the conceptual model**. Explicitly describe consistencies and metaphors in the user documentation to assist learning. The help system should aim to improve mental models.
- **Promote the development of both novice and expert mental models**. Protect novices from system complexity by employing progressive disclosure of functionality.

Additionally, a conceptual model describes how the interface will present objects, their relationships, properties, and the actions performed. Base the conceptual model on the user's mental model. The goal is to facilitate the development of a useful mental model of the system for the user.

4 c) Discuss about the usage of objects and metaphors in design.

In user interface design, **objects** and **metaphors** play important roles in creating a system that is both intuitive and easy to learn

Objects:

• A graphical system is comprised of objects and actions
• Objects are what people see on the screen and are manipulated as a single unit.
• A well-designed system keeps users focused on objects rather than on how to carry out actions.
• IBM's Common User Access application breaks objects into three meaningful classes: data, container, and device.
    ◦ **Data objects** present information.
    ◦ **Container objects** are collections of other objects.
    ◦ **Device objects** represent physical devices.
• Microsoft Windows specifies that objects can exist within the context of other objects, where one object may affect how another appears or behaves. These relationships are called collections, constraints, composites and containers.

◦ A **collection** is the simplest relationship where objects share a common aspect.

When defining objects in the design process, it is important to:

- Determine all objects that have to be manipulated to get work done.
- Describe the objects used in tasks.
- Establish object behaviour and characteristics that differentiate each kind of object.
- Determine the relationship of objects to each other and the people using them.
- Describe the actions performed and the objects to which actions apply.
- Define information or attributes that each object in the task must preserve, display or allow to be edited.
- Identify the objects and actions that appear most often in the workflow.
- Make the several most important objects very obvious and easy to manipulate.

Metaphors:

- A metaphor is a concept where one's body of knowledge about one thing is used to understand something else.
- Metaphors act as building blocks of a system, aiding understanding of how a system works and is organised.
- The success of graphical systems can be attributed, in part, to their employing the desktop metaphor.
- A common metaphor in a graphical system is the desktop and its components, such as folders and a trash can. The Web utilises a library metaphor for the activities of browsing and searching.

When developing metaphors:

- Choose the analogy that works best for each object and its actions.
- Use real-world metaphors.
- Use simple metaphors.
- Use common metaphors.
- Multiple metaphors may coexist.
- Use major metaphors, even if it can't exactly replicate them visually.
- Test the selected metaphors.

However, it is important to remember that today's technology permits doing a lot of things not even thinkable in the old manual world, and not to be constrained from developing a more powerful interface because a current metaphor just happens to exist.

In summary, both objects and metaphors contribute significantly to a user-friendly design by leveraging familiarity and facilitating intuitive interactions.

Q. 5 a) With diagrams, illustrate the structure of menus.

Any organization will have multiple applications, interfaces and utilities, specific software for carrying out their core processes and day-to-day activities. So, the organization will need to implement different kind of applications with differing menu requirements. Some of the required menu structures are described below:

Structures of Menus

Menus vary in form from very simple to very complex. They may range from small dialog boxes requesting the user to choose between one of two alternatives, to hierarchical tree schemes with many branches and level of depth. A menu's structure defines the amount of control given to the user in performing a task. The most common structures are the following.

Single Menus

In this simplest form of menu, a single screen or window is presented to seek the user's input or request an action to be performed, as illustrated in Figure shown below..

❍ Choice 1
❍ Choice 2
❍ Choice 3

A single menu may be iterative if it requires data to be entered into it and this data input is subject to a validity check that fails. The menu will then be represented to the user with a message requesting reentry of valid data.

Sequential Linear Menus

Sequential linear menus are presented on a series of screens possessing only one path. The menu screens are presented in a preset order, and, generally, their objective is for specifying parameters or for entering data. The length of the path may be short, or long, depending upon the nature of the information being collected. All the menus are important to the process at hand and must be answered in some manner by the user.

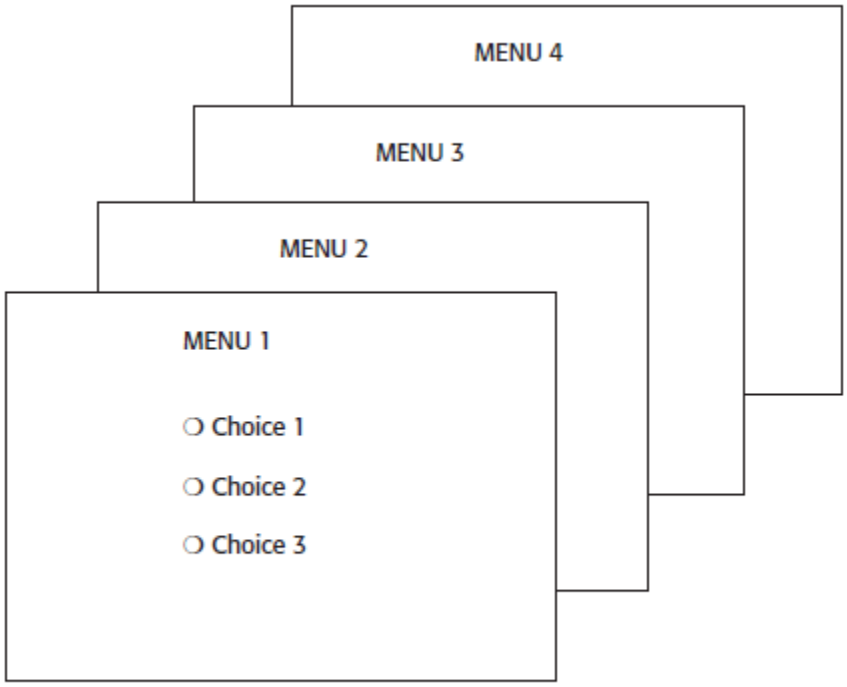


Figure 4.2 Sequential linear menus.

Simultaneous Menus

Instead of being presented on separate screens, all menu options are available simultaneously, as illustrated in Figure 4.3. The menu may be completed in the order desired by the user, choices being skipped and returned to later. All alternatives are visible for reminding of choices, comparing choices, and changing answers.

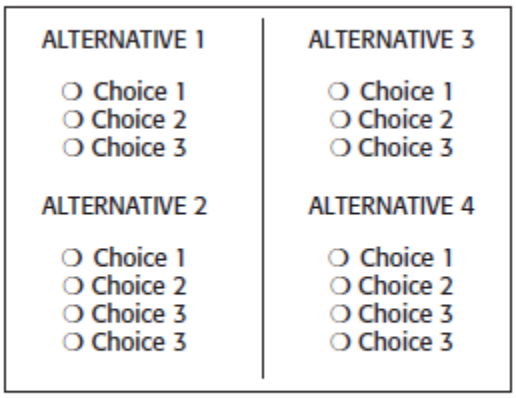


Figure 4.3 Simultaneous menus.

Hierarchical Menus

A hierarchical structure results in an increasing refinement of choice as menus are stepped through, for example, from options, to suboptions, 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. Hierarchical structures are characterized by depth and breadth, depth being the number of choice levels one must traverse to reach the destination, breadth being the number of alternatives found at each level.

A hierarchical menu is illustrated in Figure 4.4. Note that the top level of the tree is considered level 0 with subsequent levels numbered sequentially beginning with number 1. Starting at the top, level 0, two selections, or mouse clicks, are required to reach level 2.

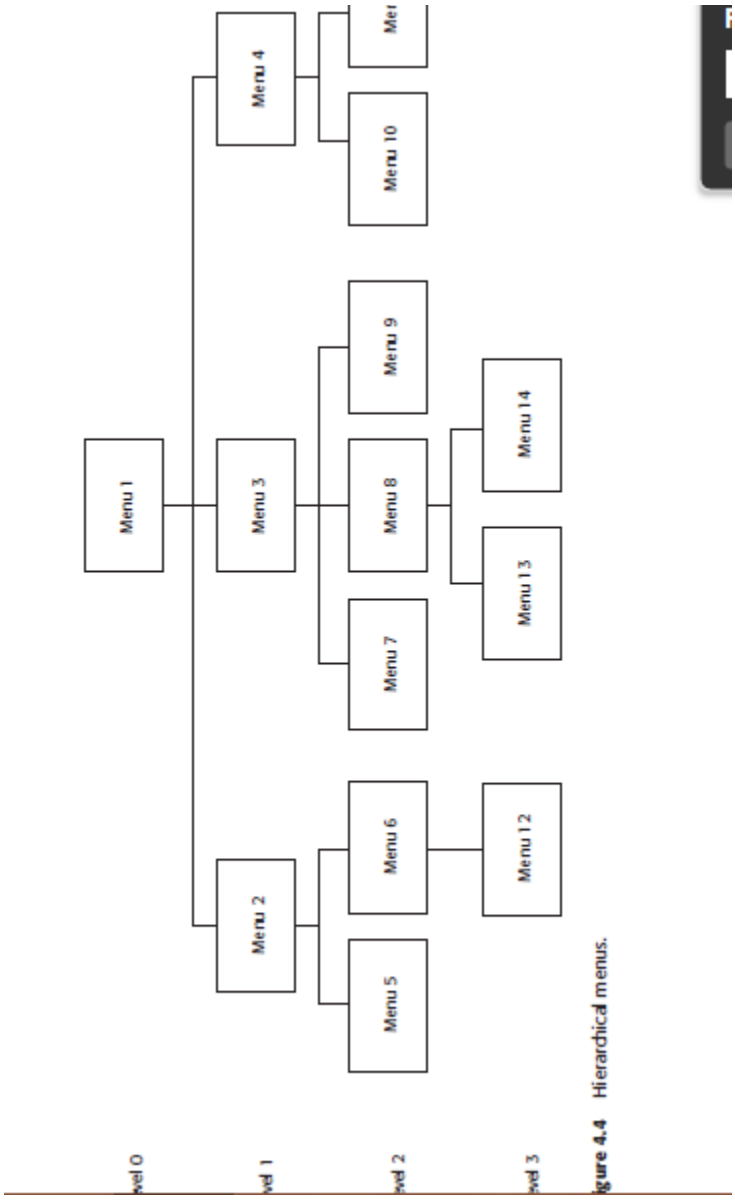


Figure 4.4 Hierarchical menus.

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. From the user's perspective there is no top-down traversal of the menu system but an almost unhindered wandering between any two menus of interest. A connected menu system may be cyclical, with movement permitted in either direction between menus, or a cyclical, with movement permitted in only one direction. These menus also vary in connectivity, the extent to
which menus are linked by multiple paths. (In a hierarchical menu system, the ability to go back to a previous menu or to return to the top-level menu are also examples, although restricted, of connected menus.)
The biggest advantage of a connected menu network is that it gives the user full control over the navigation flow. Its disadvantage is its complexity, and its navigation,may be daunting for an inexperienced user. An example connected menu structure is represented in Figure 4.5.

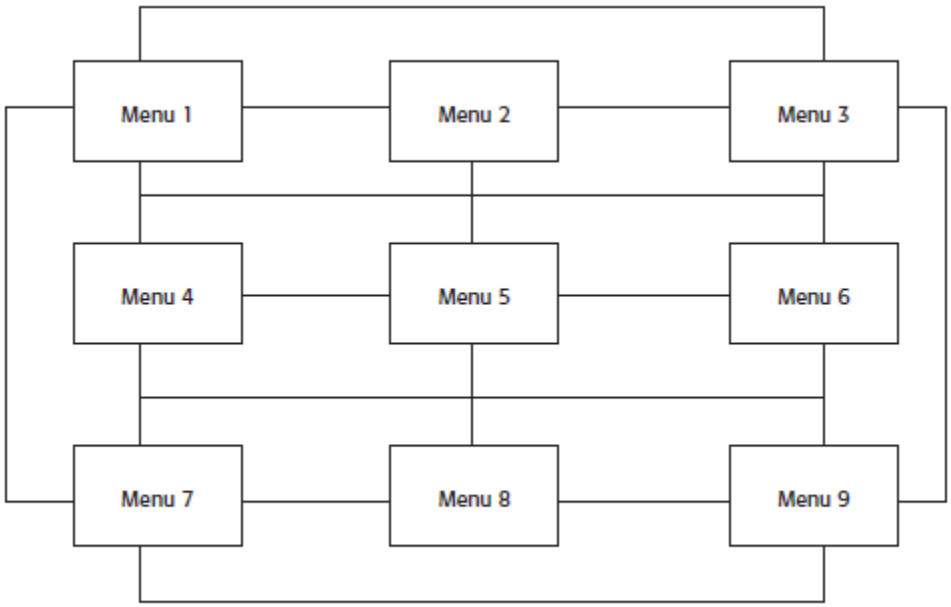


Figure 4.5 Connected menus.

Event-Trapping Menus

Event Trapping menus provide an ever-present background of control over the system's state and parameters while the user is working on a foreground task. They are, in essence, a set of simultaneous menus imposed on hierarchical menus. In a graphical system, for example, existing together are a simultaneous menu, the menu bar, and a hierarchy—the menu bar and its pull-downs. Event-trapping menus generally serve one of three functions. (1) They may immediately change some parameter in the current environment (bold a piece of text), (2) they may take the user out of the current environment to perform a function without leaving the current environment (perform a spell check), or (3) they may exit the current environment and allow the user to move to a totally new environment (Exit).

Q. 5 b) Describe different functions performed by the menus.

Menus can perform several functions from the user's perspective:

- **Navigation to a new menu** Each selection guides the user towards a goal through a hierarchical menu, but errors can lead to wrong paths.
- **Execute an action or procedure** Selecting an option directs the computer to perform an action, such as opening or closing a file, with potential consequences for errors depending on the action's nature.
- **Displaying information** The primary goal may be to present information, like searching a database or browsing the web, where the user focuses on the information and the content guides their path.
- **Data or parameter input** Each selection specifies input data or a parameter value, with the user focused on providing information, and errors can be corrected if detected.

Q. 5 c) Discuss about the content of menus.

A menu comprises four key elements: its **context**, **title**, **choice descriptions**, and **completion instructions**.

- **Menu Context:** Provides information to keep the user oriented, which is especially important in complex or hierarchical menu systems where users can easily lose their position. Navigation feedback can be provided through verbal or spatial linkage.

    - Verbal linkage lists choices made on previous menus to guide the user. The title should reflect the option selected on the previous menu, and its content should reflect its title.
    - Spatial linkage uses graphic methods, such as overlapping succeeding menu screens, to show a series of choices in a single view to easily ascertain progress and distance.
- **Menu Title:** The title provides context for the current set of choices and must reflect the choice selected on the previously displayed menu. The title should orient the viewer to the menu's content and purpose. Main menu titles should be short, clear, distinctive, and descriptive. Submenu titles must be worded exactly the same as the menu choice previously selected to display them.
- **Choice Descriptions:** These are the alternatives available to the user, ranging from mnemonic, numeric, or alphabetised listings to single words, phrases, or full sentences. The style reflects the user's experience, the nature of the choices, the selection mechanism, and the system. Menu item descriptions should be meaningful, familiar, fully spelled out, concise, and distinctive. Descriptions may be single words, compound words, or multiple words or phrases. Main menu bar items should be a single word, if possible.
- **Completion Instructions:** Completion instructions may be included on a menu, but these instructions should be easily ignored by those who do not need them. Explicit menu completion instructions can be provided for novice or inexperienced menu users. These should be placed in a position just preceding the part, or parts, of the menu to which they apply and presented in a mixed-case, sentence-style font.

Q. 6 a) **List all graphical menus and explain any one in detail**

The sources describe several types of graphical menus. These include:

- **Menu Bar:** A horizontal row of descriptions at the top of a window that serve as headings for actions in associated pull-down menus.
- **Pull-Down Menu:** A vertically arrayed listing of choices that appears when a menu bar item is selected.
- **Cascading Menus:** Lower-level menus that appear when an item on a pull-down menu is selected.
- **Pop-Up Menus:** Menus that present alternatives or choices within the context of a task.
- **Tear-Off Menus:** Menus that can be detached from their original location and moved elsewhere on the screen.
- **Iconic Menus:** Menus that portray items or objects in a graphic or pictorial form.
- **Pie Menus:** This menu type is also mentioned in the sources.

Here's a more detailed explanation of **Pull-Down Menus**:

Proper Usage: Pull-down menus are used to initiate frequently used application actions that take place on a wide variety of different windows. They are best suited for a small number of items that are represented textually and whose content rarely changes.

Display: A pull-down menu should display all possible alternatives, with items that cannot be chosen due to the current state of the application grayed out or dimmed. If all items are not applicable, they must still be capable of being retrieved for perusal through the menu bar.

Size: A pull-down menu must contain a minimum of two choices and should be restricted to no more than five to ten choices, preferably eight or less.

Structure: To simplify a higher-level menu and provide easier browsing. Restrict to one–two cascades.

Organization: Frequent or critical items should be placed at the top, and related choices should be grouped together. A traditional, split, or folded structure can be used as necessary. Choices found on more than one pull-down should be

consistently positioned. Multicolumn menus are not desirable; if necessary, organize pull-downs from top-to-bottom, then left-to-right.

Q. 6 b) **Illustrate the concept of formatting the menus**

Formatting of Menus

What follows is 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.

— Critical options should be continuously displayed, however.

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.

• 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.

o 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.

• Easier hiding of inappropriate choices.

• Less likelihood of confusing similar choices since there is less likelihood that they will be seen together.

• Greater depth disadvantages are:

• More steps and longer time to reach one's objective.

• More difficulties in learning since relationships between elements cannot always be seen.

• More difficulties in predicting what lies below, resulting in increased likelihood of going down wrong paths or getting lost.

• Higher error rates.

Complexity

• Provide both simple and complex menus.

• Simple: a minimal set of actions and menus.

• Complex: a complete set of actions and menus.

Item Arrangement

• Align alternatives or choices into single columns whenever possible.

— Orient for top-to-bottom reading.

— Left-justify descriptions.

.

Q. 6 c) **Describe any 3 guidelines in the phrasing of menus**

Phrasing the Menu

• A menu must communicate to the user information about:

o The nature and purpose of the menu itself.

o The nature and purpose of each presented choice.

o 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.

o 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.

Menu Instructions

• For novice or inexperienced users, provide menu completion instructions.

— Place the instructionsin 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

Q. 7 a) **With diagrams, illustrate the different components of a window.**

Components of a Window

A typical window may be composed of up to a dozen or so elements. Some appear on all windows; others only on certain kinds of windows, or under certain conditions. For consistency purposes, these elements should always be located in the same position within a window.

Figure 5.1 Microsoft Windows primary window.

Window components are

1. Frame

A window will have a frame or border, usually rectangular in shape, to define its boundaries
and distinguish it from other windows. While a border need not be
rectangular, this shape is a preferred shape for most people. Also, textual materials, which are usually read from left to right, fit most efficiently within this structure. The border comprises a line of variable thickness and color.

2. Title Bar

The title bar is the top edge of the window, inside its border and extending its entire width. This title bar is also referred to by some platforms as the *caption*, *caption bar*, or *title area*. The title bar contains a descriptive title identifying the purpose or content of the window.

3. Title Bar Icon

Located at the left corner of the title bar in a primary window, this button is used in Windows to retrieve a pull-down menu of commands that apply to the object in the window. It is 16 × 16 version of the icon of the object being viewed. When clicked with the secondary mouse button, the commands applying to the object are presented. Microsoft suggests that:

- If the window contains a tool or utility (that is, an application that does not create,

load, and save its own data files), a small version of the application's icon should be placed there instead.

- If the application creates, loads, and saves documents or data files and the window

represents the view of one of its files, a small version of the icon that represents its document or data file type should be placed there. Even if the user has not yet saved the file, display the data file icon rather than the application icon, and again display the data file icon after the user saves the file.

4. Window Sizing Buttons

A window's title bar must have equivalent commands on the pop-up or shortcut menu for that window. When these buttons are displayed, use the following guidelines:

When a window does not support a command, do not display its command button.

The *Close* button always appears as the rightmost button. Leave a gap between it and any other buttons. The *Minimize* button always precedes the *Maximize* button. The *Restore* button always replaces the *Maximize* button or the *Minimize* button when that command is carried out.

5. What's This? Button

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.

Figure 5.2 What's This? button.

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. A system will typically provide a default set of menu actions that can be
augmented by an application. In the past, some platforms have called the menu bar an
action bar.

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 area, extending its entire length. Horizontal scrolling is accomplished through a scroll bar located at the bottom of the work area.

9. Split Box

A window can be split into two or more pieces or panes by manipulating a *split box* located
above a vertical scroll bar or to the left of a horizontal scroll bar. 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*. Splitting a window permits multiple views of an object. A split window allows the user to: Examine two parts of a document at the same time.
Display different, yet simultaneous, views of the same information.

10. Toolbar

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*. Each toolbar band includes a single-grip handle to enable the user to resize or rearrange the toolbars. When the user moves the pointer over the grip, it changes to a two-headed arrow. When the user drags the grip, the pointer changes to a split move pointer.

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

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

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.

Q. 7 b) Provide various advantages of using windows

The advantages of using windows in user interface design are numerous, as they closely mirror real-world desk workspaces. Here's a breakdown of these advantages:

- **Reduced memory load:** Windows act as external memories, reducing the burden on short-term memory.
- **Easy task switching:** They facilitate easier switching between tasks and maintaining context, eliminating the need to reestablish one's place repeatedly.
- **Access to more information:** Windows provide access to more information than a single display by layering information and presenting important details on top.
- **Different levels of information:** Information can be examined in increasing levels of detail through the use of multiple windows.
- **Sequential presentation:** Steps to accomplish a task can be presented sequentially, which is especially useful when the information-collection process leads down various paths.
- **Combining multiple sources:** Windows allow text from several documents to be reviewed and combined into one.
- **Performing multiple tasks:** More than one task can be performed at one time.
- **Multiple representations of the same task:** The same thing can be looked at in several ways.

Different window presentation styles (tiled, overlapping, and cascading) also offer unique advantages:

- **Tiled windows:** These are considered less complex, easier for novice users, and better for tasks requiring little window manipulation. They ensure that open

windows are always visible and completely seen, eliminating the possibility of them being lost or hidden.
- **Overlapping windows:** These resemble a familiar desktop, offer greater user control, and allow windows to maintain larger and consistent sizes and positions. They are better for switching between tasks and for expert users.
- **Cascading windows:** These ensure that no window is ever completely hidden and make bringing any window to the front easier.

In graphical systems, windows have been found to simplify interfaces, reduce memory demands on users, use information-processing capabilities more effectively, and dramatically reduce system learning requirements.

Q. 7 c) Explain the characteristics of window

Windows in a graphical user interface (GUI) possess several key characteristics that define their functionality and how users interact with them. These characteristics can be categorised as follows:

- **Name or Title:** A window has a name or title that allows it to be identified. The title should clearly and concisely describe the purpose of the window.
- **Size:** Windows come with default height and width, which can often be adjusted.
- **State:** A window can be accessible (active) or not accessible. Only active windows can have their contents altered.
- **Visibility:** This refers to the portion of the window that can be seen. A window may be partially or fully hidden behind another window, or the information within it may extend beyond the display area.
- **Location:** A window has a location relative to the display boundary.
- **Presentation:** This is the arrangement of the window in relation to other windows, such as tiled, overlapping, or cascading.
- **Management Capabilities:** These are the methods for manipulating the window on the screen.
- **Highlight:** This is the part of the window that is selected.
- **Function:** This refers to the task or application to which the window is dedicated.

In addition to these general characteristics, windows also have defining concepts in a graphical system:

- **Sophisticated Visual Presentation:** GUIs use visual elements like windows, icons, and text in multiple fonts, colours, and sizes. They may also include animation and video.
- **Pick-and-Click Interaction:** Users interact by selecting elements on the screen, often with a mouse.
- **Restricted Set of Interface Options:** Actions and choices are presented in a limited, but manageable set for the user.
- **Visualisation:** Complex data can be presented graphically to improve user understanding.
- **Object Orientation:** Systems are made up of objects (data, containers, devices) that users manipulate. Objects can exist within the context of other objects, influencing their appearance or behaviour through relationships like collections, constraints, composites and containers.
- **Use of Recognition Memory:** Users rely on recognising items displayed, which is easier than recall. Lists of options are preferred over command line interfaces.

- **Concurrent Performance of Functions:** GUIs can support multiple programs running at the same time.

Another important object characteristic is **persistence**, which is the maintenance of a state once it is established. An object's state (e.g., window size, cursor location, scroll position) should always be automatically preserved when the user changes it. Objects also have **properties or attributes**, which are unique characteristics that can be changed by users. Examples include text styles, font sizes, or window background colours.

Q. 8 a) **Discuss about different presentation styles of a window.**

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.

I) Tiled Windows

Figure 5.4 Tiled windows.

Tiled windows, the first and oldest kind of window, are felt to have these **advantages**:

- The system usually allocates and positions windows for the user, eliminating the necessity to make positioning decisions.
- Open windows are always visible, eliminating the possibility of them being lost and forgotten.
- Every window is always completely visible, eliminating the possibility of information being hidden.
- They are perceived as less complex than overlapping windows, possibly because there are fewer management operations or they seem less "magical."
- They are easier, according to studies, for novice or inexperienced people to learn and use.
- They yield better user performance for tasks where the data requires little window manipulation to complete the task.

Perceived **disadvantages** include the following:

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

II) Overlapping Windows

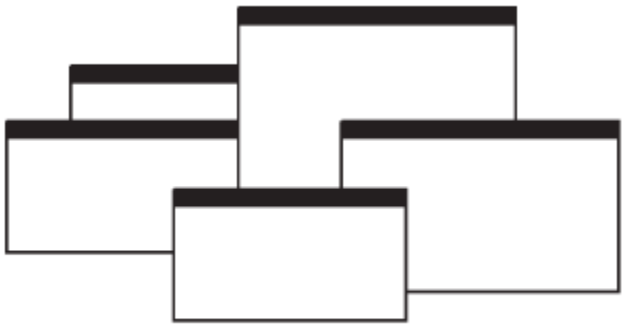


Figure 5.5 Overlapping windows.

Overlapping windows, illustrated in Figure 5.5, 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.
They have the following **advantages**:

- Visually, their look is three-dimensional, resembling the desktop that is familiar to the user.
- Greater control allows the user to organize the windows to meet his or her needs.
- Windows can maintain larger sizes.
- Windows can maintain consistent sizes.
- Windows can maintain consistent positions.
- Screen space conservation is not a problem, because windows can be placed on top of one another.
- There is less pressure to close or delete windows no longer needed.
- The possibility exists for less visual crowding and complexity. Larger borders can be maintained around window information, and the window is more clearly set off against its background. Windows can also be expanded to fill the entire display.
- They yield better user performance for tasks where the data requires much window manipulation to complete the task.

Disadvantages include the following:

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

III) Cascading Windows

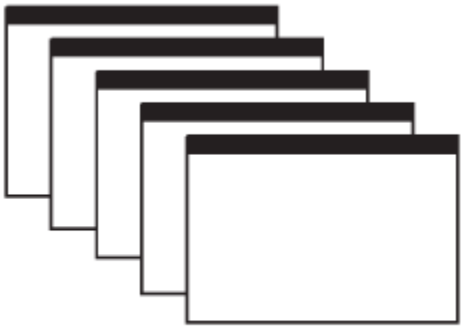


Figure 5.6 Cascading windows.

A special type of overlapping window has the windows automatically arranged in a regular progression.
Advantages of this approach include the following:

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

IV) 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.

Tiled windows. Tiled windows seem to be better for single-task activities and data that must be seen simultaneously. A study found that tasks requiring little window manipulation were carried out faster using tiled windows. They also found that novice users performed better with tiled windows, regardless of the task.

Overlapping windows. Overlapping windows seem to be better for situations that necessitate switching between tasks. A research study concluded that tasks requiring much window manipulation could be performed faster with overlapping windows but only if user window expertise existed. For novice users, tasks requiring much window manipulation were carried out faster with tiled windows. Therefore, the advantage to overlapping windows comes only after a certain level of expertise is achieved. Overlapping windows are the preferred presentation scheme.

Window Management

Microsoft Windows also provides several window management schemes,

1. A *single document interface*-A single primary window with a set of secondary windows.
2. *A multiple-document interface*-— 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.

The purpose of this scheme of windows is to provide multiple views of the same object, to permit comparisons
among related objects, and to present multiple parts of an application

3. *Workbooks*-Tabs are used as a navigational interface to move between different sections.
4. *Projects*- A technique that consists of a container: a project window holding a set of objects.

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.

Q. 8 b) **Provide the guidelines for selecting the device based controls.**

To select the proper device-based controls, consider the characteristics of the task, user, environment, and hardware, as well as the device in relation to the application. Flexibility is also key.

Here's a breakdown of guidelines for selecting the proper input device:

- **Task characteristics**:
    - **Keyboards** are suited for tasks involving heavy text entry and manipulation, and for movement through structured arrays consisting of a few discrete objects.
    - Provide an alternative **pointing device** for graphical or drawing tasks.
        - **Mouse**: Pointing, selecting, drawing, and dragging.
        - **Joystick**: Selecting and tracking.
        - **Trackball**: Pointing, selecting, and tracking.
        - **Touch screen**: Pointing and selecting.
        - **Graphic tablet**: Pointing, selecting, drawing, and dragging.
    - Provide **touch screens** under the following conditions:
        - Minimal training opportunity.
        - Targets are large, discrete, and spread out.
        - Low frequency of use.
        - Desk space is limited.
        - Little to no text input requirement.
- **User characteristics and preferences**:
    - Provide **keyboards** for touch typists.
    - Consider whether the user can easily and comfortably operate the control, and whether they are familiar with the standard keyboard.
    - Consider device-learning requirements. **Direct pointing devices** (light pen, stylus, and touch screen) are intuitive and easy to learn, and are best in situations where people cannot be expected to spend time learning, and for infrequent system users.
- **Environmental characteristics**: Consider whether the device fits easily into the work environment. Public access systems will require a sturdy device such as a touch screen.
- **Hardware characteristics**: Ensure the device is of a quality that permits easy performance of all the necessary tasks.
- **Device in relation to the application**: Ensure the device is satisfactory for the application.
- **Flexibility**: Providing more than one kind of device will give the user choices in how to most efficiently accomplish whatever tasks must be performed. A keyboard paired with another kind of pointing device is almost always necessary.
- **Eye and hand movements**: Minimise eye and hand movements between devices.
- **Versatility and Text Entry**: Keyboards are versatile and useful for text entry tasks.

Selecting the right device-based control is crucial for system success. A good fit between user and control leads to fast, accurate performance, while a poor fit results in lower productivity, more errors, and increased user fatigue.

Q. 8 c) **Explain the characteristics of touch screen and keyboard.**

Here's an explanation of the characteristics of touch screens and keyboards:

Touch Screen:

- **Description**: A special screen surface that is sensitive to touch by a finger or stylus.
- **Direct Relationship**: Offers a direct relationship between hand and pointer location in terms of direction, distance, and speed. This directness simplifies the psychomotor skills needed for use.
- **Direct Manipulation**: The control (finger or stylus) is on the same plane as the pointer, making it a direct pointing device.

- **Space Efficiency**: Requires no additional desk space.
- **Durability**: Stands up well in high-use environments.
- **Accuracy**: A stylus is generally more accurate than using a finger.
- **Visibility Issues**: A finger may obscure part of the screen.
- **Target Size**: Finger use may be inaccurate with small objects. Objects should be at least 3/4 x 3/4 inches in size with a separation of at least 1/8 inch.
- **Fatigue**: Can be fatiguing to use for extended periods.
- **Screen Maintenance**: The screen may be soiled or damaged.
- **Hand Movement**: Requires moving the hand far from the keyboard.
- **Feedback**: Should provide visual feedback in response to activation. Auditory feedback may also be appropriate.
- **Confirmation**: When consequences are destructive, require confirmation after selection to eliminate inadvertent selection.
- **Instructional Invitation**: Provide an instructional invitation to begin using.

Keyboard:

- **Description**: A standard typewriter keyboard with cursor movement keys.
- **Familiarity**: It is a familiar device.
- **Accuracy**: Keyboards are accurate.
- **Space Efficiency**: It does not take up additional desk space.
- **Text and Data Entry**: Very useful for entering and editing text and alphanumeric data.
- **Shortcuts**: Allows keyed shortcuts (accelerators) and keyboard mnemonics (equivalents).
- **Versatility**: Advantageous for performing actions when fewer mouse buttons exist or with very large screens.
- **Touch Typists**: Efficient for touch typists.
- **Discrete Input**: Requires discrete finger actions to operate.
- **Indirect Relationship**: No direct relationship exists between finger or hand movement on the keys and cursor movement on screen in terms of speed and distance.
- **Non-Touch Typists**: It can be slow for non-touch-typists.
- **Complexity**: Keyboards can be over-elaborate.
- **Pointing Inefficiency**: Slower than other devices in pointing.
- **Keyboard Accelerators**: Provide keyboard accelerators.
- **Keyboard Equivalents**: Provide keyboard equivalents.

In summary, **touch screens** excel in directness and simplicity, making them suitable for environments with minimal training and limited desk space. However, they can be fatiguing and less accurate for small targets. **Keyboards**, on the other hand, offer accuracy and efficiency for text-based tasks, particularly for touch typists, but they lack the direct manipulation advantages of touch screens.

Q. 9 a) **What are operable controls? Elaborate the usage of command buttons with guidelines.**

Operable controls are screen-based elements that allow users to input, select, change, or edit data, or to execute commands. They are manipulable, changeable, or settable components of a user interface. Classes of operable controls include buttons, text

entry/read-only fields, selection controls, combination entry/selection controls, and other specialised controls.

Command Buttons

Command buttons are a primary type of operable control. They are square or rectangular controls containing a label (text, graphics, or both) that indicates the action to be performed when the button is selected.

- **Purpose**: Command buttons are used to start actions, change properties, or display a pop-up menu. They can cause something to happen immediately, display another window, display a menu of options, or set a mode or property value. In web design, buttons should only perform actions and not retrieve or display information.
- **Advantages**: Command buttons are always visible, reminding users of available choices. They are conveniently located and can provide meaningful descriptions of their functions. Their size makes them easy to select.
- **Proper Usage**: Command buttons are best for frequently used actions within a window. They can be the only command method available in some windows. For windows with a menu bar, command buttons offer fast access to frequently used or critical commands. For windows without a menu bar, they provide access to all necessary commands.

Guidelines for Command Buttons

- **Structure**: Use a rectangular shape with a label inside and give the button a raised appearance. Maintain consistency in style throughout an application.
- **Labels**:
    - Use standard button labels when available and provide meaningful descriptions of the actions to be performed.
    - Use single-word labels whenever possible, but use two to three words for clarity if necessary.
    - Use mixed-case letters with the first letter of each significant label word capitalised.
    - Centre the label within the button borders, leaving at least two pixels between the text and the button border.
- **Size**: Provide as large a button as feasible, maintaining consistent button heights and widths. The minimum height should be 25 pixels. Expand the button's size to fit the label properly, and do not reduce the font size.
- **Number**: Restrict the number of buttons on a window to six or fewer.
- **Location and Layout**:
    - Position command buttons consistently within a window to enable memorisation of button locations.
    - Allocate a space for buttons before establishing other control locations.
    - If a button has a contingent relationship to another control, position it adjacent to the related control in the order in which the controls are usually operated.
    - For web pages longer than one screen, repeat the buttons at the top and bottom of the page.
    - For exiting and expanding/invoking feature buttons, do not align with other screen controls or present them within a line border.
    - Provide equal and adequate spacing between adjacent buttons and between buttons and the screen body controls.
- **Organization**:

    - Organise standard buttons in the manner recommended by the platform being used.
    - For other buttons, organise them in common and customary grouping schemes.
    - For buttons ordered left to right or top to bottom, place those for most frequent actions to the left or top, respectively.
    - Keep related buttons grouped together and separate potentially destructive buttons from frequently chosen selections.
    - Buttons found on more than one window should be consistently positioned.
    - For mutually exclusive actions, use two buttons; do not dynamically change the text.
- **Intent Indicators**:
    - When a button causes an action to be performed immediately, no intent indicator is necessary.
    - When a button leads to a cascading dialogue, include an ellipsis (...) after the label.
    - When a button leads to a menu, include a triangle pointing in the direction the menu will appear after the label.
    - When a button leads to an expanding dialogue, include a double arrow (>>) with the label.
    - When a button has a contingent relationship to another control that must be indicated, include a single arrow (->) pointing at the control.
- **Expansion Buttons**: Gray out a button after expansion and provide a contraction button, if necessary, located beneath or to the right of the expansion button.
- **Unavailable Choices**: Temporarily unavailable choices should be dimmed or grayed out.
- **Keyboard Equivalents and Accelerators**: Assign a keyboard equivalent mnemonic to each button to facilitate keyboard selection. Keyboard accelerators may also be assigned to buttons to facilitate keyboard activation.
- **Scrolling**: If a window can be scrolled, do not scroll the command buttons, unless the screen cannot scroll independently of the buttons. Use buttons, not scroll bars, to move between multi-page forms, labelling the buttons "Next" and "Previous".
- **Button Activation**: Highlight the button in some visually distinctive manner when the pointer is resting on it and when it has been activated or pressed.

By following these guidelines, command buttons can be effectively used to create an intuitive and efficient user interface.

Q. 9 b) **Discuss about tool tips and progress indicator controls.**

ToolTips

- **Description**: A ToolTip, also called a ScreenTip, is a **small pop-up window** containing **descriptive text** that appears when a pointer is moved over a control or element. ToolTips may be used when the control or element either does not have a label or requires additional descriptive or status information.
- **Purpose**: The primary purpose of a ToolTip is to **provide descriptive information** about a control or screen element. It is useful for identifying controls that lack captions or have abbreviated captions.

Advantages:

- Identifies otherwise unidentified controls.
- Reduces screen clutter by reducing the need for control captions and descriptive information.
- Enables control size reduction.

Disadvantages:

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

Proper Usage:

- To identify a control without a caption.
- To provide additional descriptive or status information about a screen element.

ToolTip Guidelines:

- **Display**: Display after a short time-out.
- **Labels**: For toolbars, provide a brief word as a label in mixed case, headline style with no ending punctuation. For other elements, provide a brief phrase presenting descriptive or status information in mixed case, sentence style.
- **Placement**: Present ToolTips at the lower-right edge of the pointer, ensuring they are fully displayed on the screen. For text boxes, display ToolTips centred under the control.
- **Colour**: Display them in the standard system ToolTip colours.
- **Removal**: Remove the ToolTip when the control is activated or the pointer is moved away.
- **Good Design**: Don't substitute ToolTips for good design. Screen elements should be designed for maximum comprehension, and ToolTips should be supplements.
- Make application-specific ToolTips consistent with system-supplied ToolTips, including using the system's colour setting to distinguish them.

Progress Indicators

- **Description**: A progress indicator is a **rectangular bar** that fills as a process is being performed. The filled-in area indicates the **percentage** of the process that has been completed. It is sometimes called a progress bar.
- **Purpose**: To provide **feedback** concerning the completion of a lengthy operation.
- **Proper Usage**: To provide an indication of the proportion of a process completed.

Progress Indicator Guidelines:

- **Filling**: If horizontally arrayed, fill it from left to right; if vertically arrayed, fill it from bottom to top.
- **Colour**: Fill it with a colour or a shade of gray.
- **Text**: Include descriptive text for the process, as necessary, and place the text outside of the control.

Progress indicators are also discussed in the context of providing feedback during time delays.

Q. 9 c) Briefly **explain Radio Button control.**

Radio Buttons
■ 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.
• Not easily remembered.
• In need of a textual description to meaningfully describe the alternatives.
• Most easily understood when the alternatives can be seen together and compared to one another.
• Never changed in content.
— Do not use:
• For commands.
• Singly to indicate the presence or absence of a state.

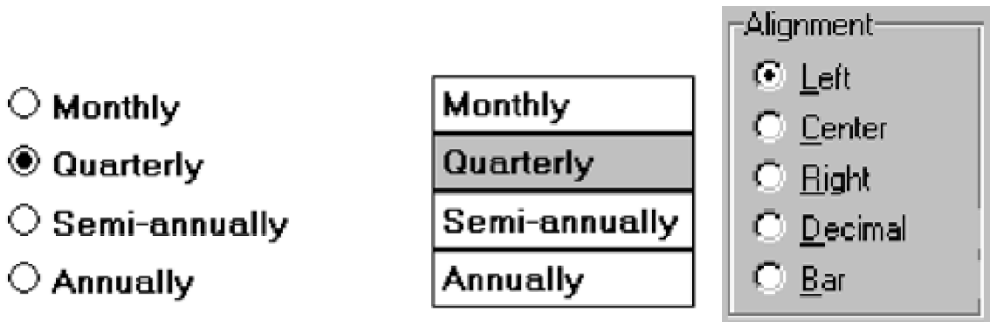


Fig : Radio Buttons

Choice Descriptions
■ Provide meaningful, fully spelled-out choice descriptions clearly describing the values or effects set by the radio buttons.
■ Display in a single line of text.
■ Display using mixed-case letters, using the sentence style.
■ Position descriptions to the right of the button. Separate them by at least one space from the button.
■ When a choice is conditionally unavailable for selection, display the choice description grayed out or dimmed.
■ Include a None choice if it adds clarity.

Size
■ Show a minimum of two choices, a maximum of eight.

Defaults
■ When the control possesses a state or affect that has been predetermined to have a higher probability of selection than the others, designate it as the default and display its button filled in.
■ When the control includes choices whose states cannot be predetermined, display all the buttons without setting a dot, or in the *indeterminate* state.

■ When a multiple selection includes choices whose states vary, display the buttons in another unique manner, or in the *mixed value* state.

Structure

■ A columnar orientation is the preferred manner of presentation.

■ Left-align the buttons and choice descriptions.

○ Red
◉ Yellow
○ Green
○ Blue

■ If vertical space on the screen is limited, orient the buttons horizontally.

■ Provide adequate separation between choices so that the buttons are associated with the proper description.

— A distance equal to three spaces is usually sufficient.

◉ Green ○ Blue ○ Yellow ○ Red

■ Enclose the buttons in a border to visually strengthen the relationship they possess.

○ Red
○ Yellow
○ Green
○ Blue

○ Green ○ Blue ○ Yellow ○ Red

Plan Choice: ○ Limited ○ Basic ○ Superior ○ Premium

Plan Choice: Limited Basic Superior Premium
○ ○ ○ ○

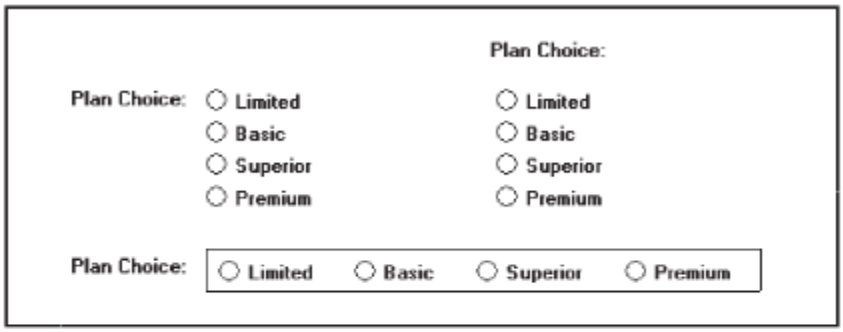
Plan Choice: ○ Limited ○ Basic ○ Superior ○ Premium

Plan Choice: Limited ○ Basic ○ Superior ○ Premium ○

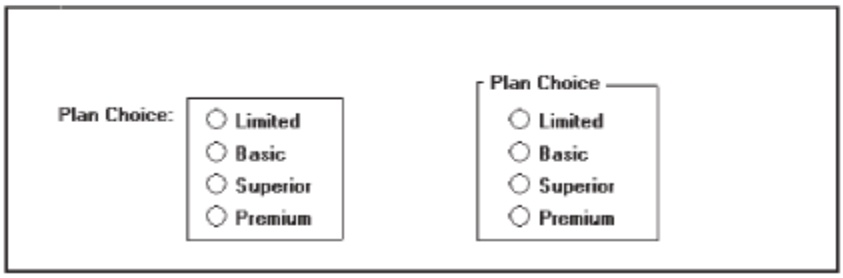
Poor

Plan Choice: ○ Limited ○ Basic ○ Superior ○ Premium

Better



Still Better



Best

Fig : Ways to, and not to, present radio buttons.

Organization

■ Arrange selections in expected order or follow other patterns such as frequency of occurrence, sequence of use, or importance.

— For selections arrayed top to bottom, begin ordering at the top.

— For selections arrayed left to right, begin ordering at the left.

■ If, under certain conditions, a choice is not available, display it subdued or less brightly than the available choices.

Related Control

■ Position any control related to a radio button immediately to the right of the choice description.

■ If the radio button choice description also acts as the label for the control that follows it, end the label with an arrow (>).

Captions

■ Structure:

— Provide a caption for each radio button control.

• Exception: In screens containing only one radio button control, the screen title may serve as the caption.

■ Display:

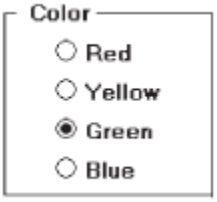
— Fully spelled out.

— In mixed-case letters, capitalizing the first letter of all significant words.

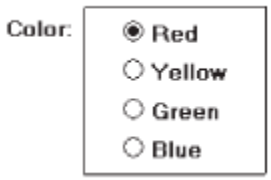
■ Columnar orientation:

— With a control border, position the caption:

• Upper-left-justified within the border.



• Alternately, the caption may be located to the left of the topmost choice description.



— Without an enclosing control border, position the caption:
• Left-justified above the choice descriptions, separated by one space line.

• Alternately, the caption may be located to the left of the topmost choice description.

■ Horizontal orientation:
— Position the caption to the left of the choice descriptions.

• Alternately, with an enclosing control border, left-justified within the border.

Keyboard Equivalents
■ Assign a keyboard mnemonic to each choice description.
■ Designate the mnemonic by underlining the applicable letter in the choice description.

Assign unique keyboard mnemonics for each alternative in the standard way, choosing the first letter (or another) and designating it by character underlining.

Selection Method and Indication
■ Pointing:
— The selection target area should be as large as possible.
• Include the button and the choice description text.
— Highlight the selection choice in some visually distinctive way when the cursor's resting on it and the choice is available for selection.
• This cursor should be as long as the longest choice description plus one space at each end. Do not place the cursor over the small button.

■ Activation:
— When a choice is selected, distinguish it visually from the unselected choices.
• A radio button should be filled in with a solid dark dot or made to look depressed or higher through use of a shadow.
— When a choice is selected, any other selected choice must be deselected.
■ Defaults:
— If a radio button control is displayed that contains a choice previously selected or

a default choice, display the selected choice as set in the control.

Q. 10 a) **What is the purpose of using prototypes? Explain any two types of prototypes.**

PROTOTYPE:

A prototype is primarily a vehicle for exploration, communication, and evaluation. Its purpose is to obtain user input in design, and to provide feedback to designers. Its major function is the communicative role it plays, not accuracy or thoroughness. A prototype enables a design to be better visualized and provides insights into how the software will look and work. It also aids in defining tasks, their flow, the interface itself, and its screens.

A prototype is a simulation of an actual system that can be quickly created. A prototype may be a rough approximation, such as a simple hand-drawn sketch, or it may be interactive, allowing the user to key or select data using controls, navigate through menus, retrieve displays of data, and perform basic system functions. A prototype need not be perfectly realistic, but it must be reasonably accurate and legible. A prototype also need not be functionally complete, possessing actual files or processing data. Today, many software support tools for prototyping are available that permit the prototype to be integrated directly into the application code.

Hand Sketches and Scenarios

Description:

— Screen sketches created by hand.

— Focus is on the design, not the interface mechanics.

— A low-fidelity prototype.

■ Advantages:

— Can be used very early in the development process.

— Suited for use by entire design team.

— No large investment of time and cost.

— No programming skill needed.

— Easily portable.

— Fast to modify and iterate.

— A rough approximation often yields more substantive critical comments.

— Easier to comprehend than functional specifications.

— Can be used to define requirements.

■ Disadvantages:

— Only a rough approximation.

— Limited in providing an understanding of navigation and flow.

— A demonstration, not an exercise.

— Driven by a facilitator, not the user.

*Sketch Creation Process**

Sketch (storyboard) the screens while determining:

— The source of the screen’s information.

— The content and structure of individual screens.

— The overall order of screens and windows.

Use an erasable medium.

■ Sketch the screens needed to complete each workflow task.

■ Try out selected metaphors and change them as necessary.

■ First, storyboard common/critical/frequent scenarios.

— Follow them from beginning to end.

— Then, go back and build in exceptions.

■ Don’t get too detailed; exact control positioning is not important, just overall order

and flow.

■ Storyboard as a team, including at least one user.

■ Only develop online prototypes when everyone agrees that a complete set of screens

has been satisfactorily sketched

Interactive Paper Prototypes

Description:

— Interface components (menus, windows, and screens) constructed of common paper technologies (Post-It notes, transparencies, and so on).

— The components are manually manipulated to reflect the dynamics of the software.

— A low-fidelity prototype.

■ Advantages:

— More illustrative of program dynamics than sketches.

— Can be used to demonstrate the interaction.

— Otherwise, generally the same as for hand-drawn sketches and scenarios.

■ Disadvantages:

— Only a rough approximation.

— A demonstration, not an exercise.

— Driven by a facilitator, not the user.

— Limited usefulness for usability testing.

Programmed Facades

■ Description:

— Examples of finished dialogs and screens for some important aspects of the system.

— Created by prototyping tools.

— Medium-fidelity to high-fidelity prototypes.

■ Advantages:

— Provide a good detailed specification for writing code.

— A vehicle for data collection.

■ Disadvantages:

— May solidify the design too soon.

— May create the false expectation that the “real thing” is only a short time away.

— More expensive to develop.

— More time-consuming to create.

— Not effective for requirements gathering.

— Not all of the functions demonstrated may be used because of cost, schedule limitations,

or lack of user interest.

— Not practical for investigating more than two or three approaches.

Prototype-Oriented Languages

■ Description:

— An example of finished dialogs and screens for some important aspects of the

system.

— Created through programming languages that support the actual programming

process.

— A high-fidelity prototype.

■ Advantages:

— May include the final code.

— Otherwise, generally the same as those of programmed facades.

■ Disadvantages:

— Generally the same as for programmed facades.

Q. 10 b) **Discuss about the concept of Cognitive walkthrus.**

Cognitive Walkthroughs

Description:
— Reviews of the interface in the context of tasks users perform.
■ Advantages:
— Allow a clear evaluation of the task flow early in the design process.
— Do not require a functioning prototype.
— Low cost.
— Can be used to evaluate alternate solutions.
— Can be performed by developers.
— More structured than a heuristic evaluation.
— Useful for assessing “exploratory learning.”
■ Disadvantages:
— Tedious to perform.
— May miss inconsistencies and general and recurring problems.
■ Guidelines:

— Needed to conduct the walkthrough are:
• A general description of proposed system users and what relevant knowledge
they possess.
• A specific description of one or more core or representative tasks to be performed.
• A list of the correct actions required to complete each of the tasks.
— Review:
• Several core or representative tasks across a range of functions.
• Proposed tasks of particular concern.
— Developers must be assigned roles of:
• Scribe to record results of the action.
• Facilitator to keep the evaluation moving.
— Start with simple tasks.
— Don’t get bogged down demanding solutions.
— Limit session to 60 to 90 minutes.

Q. 10 c) **Write short notes on Usability Testing.**

Usability Test

■ Description:
— An interface evaluation under real-world or controlled conditions.
— Measures of performance are derived for specific tasks.
— Problems are identified.
■ Advantages:
— Utilizes an actual work environment.
— Identifies serious or recurring problems.
■ Disadvantages:
— High cost for establishing facility.
— Requires a test conductor with user interface expertise.
— Emphasizes first-time system usage.
— Poorly suited for detecting inconsistency problems.