Systems Analysis and Design

Chapter 7
Design

The slide is mainly adopted from:

- J. S. Valacich, J. George, Modern Systems Analysis and Design. 8th Edition, Pearson 2017.
- I. Sommerville. Software Engineering. 10th Edition, Pearson, 2016.

Last update: Dec. 03, 2018

Learning Objectives

- Explain the role of designing databases in the analysis and design of an information system.
- Explain when to use different types of file organizations to store computer files.
- Explain the process of designing forms and reports and the deliverables for their creation.
- Apply the general guidelines for formatting forms and reports (color, format text, tables, and lists).
- Explain how to assess usability and describe how variations in users, tasks, technology, and environmental characteristics influence the usability of forms and reports.
- Explain the process of designing interfaces and dialogues and the deliverables for their creation.
- Design human-computer dialogues and understand how dialogue diagramming can be used to design dialogues.
- Design graphical user interfaces.

Introduction

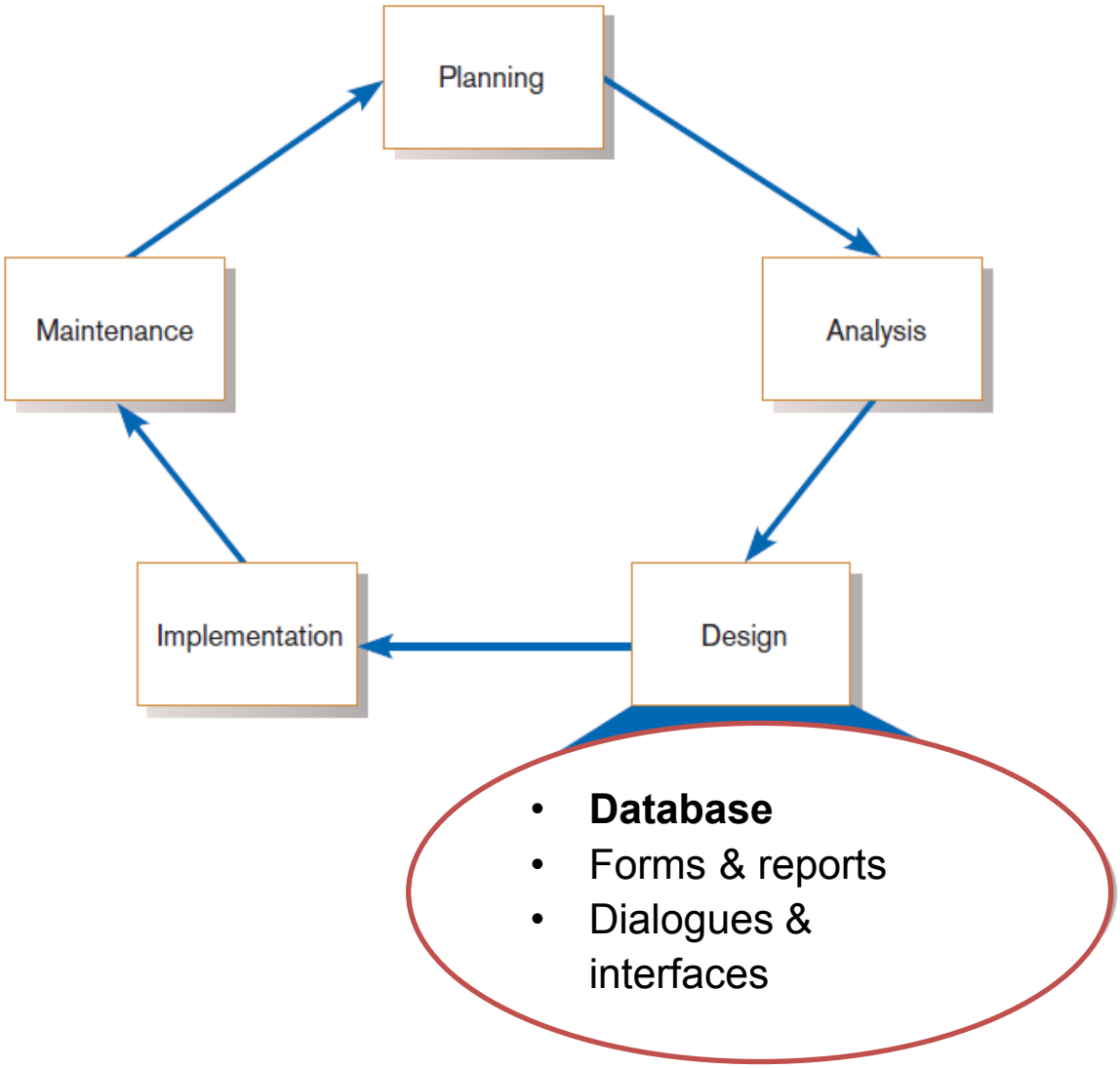


FIGURE 9-1
Systems development life cycle with design phase highlighted

Database Design

- File and database design occurs in two steps.
  1. Develop a logical database model, which describes data using notation that corresponds to a data organization used by a database management system.
     - Relational database model
  2. Prescribe the technical specifications for computer files and databases in which to store the data.
     - Physical database design provides specifications
- Logical and physical database design in parallel with other system design steps.

Logical Database Design

- Four key steps in logical database modeling and design:
  1. Develop a logical data model for each known user interface for the application using normalization principles.
  2. Combine normalized data requirements from all user interfaces into one consolidated logical database model (view integration).
  3. Translate the conceptual E-R data model for the application into normalized data requirements.
  4. Compare the consolidated logical database design with the translated E-R model and produce one final logical database model for the application.
- Deliverables and Outcomes
  - Must account for every data element on a system input or output (Normalized relations are the primary deliverable)

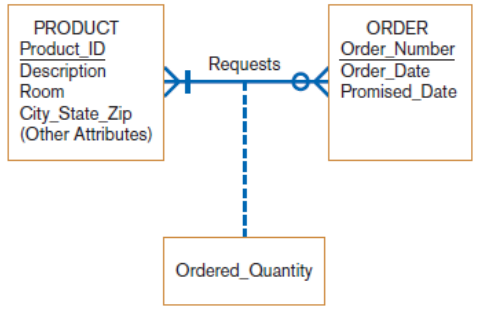
Physical Database Design

- Key physical database design decisions include:
  - Choosing a storage format for each attribute from the logical database model.
  - Grouping attributes from the logical database model into physical records.
  - Arranging related records in secondary memory (hard disks and magnetic tapes) so that records can be stored, retrieved and updated rapidly.
  - Selecting media and structures for storing data to make access more efficient.
- Deliverables and Outcomes
  - Converts relations into database tables
    - Programmers and database analysts code the definitions of the database.
    - Written in Structured Query Language (SQL)

Transforming E-R Diagrams into Relations

- It is useful to transform the conceptual data model into a set of normalized relations.
- Steps
  - *Represent entities.*
  - *Represent relationships.*
  - *Normalize the relations.*
  - *Merge the relations.*

(a)



(b) ORDER

Order_Number	Order_Date	Promised_Date
61384	2/17/2014	3/01/2014
62009	2/13/2014	2/27/2014
62807	2/15/2014	3/01/2014

ORDER LINE

Order_Number	Product_ID	Quantity_ Ordered
61384	M128	2
61384	A261	1

PRODUCT

Product_ID	Description	Room	(Other Attributes)
M128	Bookcase	Study	–
A261	Wall unit	Family	–
R149	Cabinet	Study	–

FIGURE 9-12
Representing an *M*:*N* relationship
(a) E-R diagram
(b) Relations

Physical File and Database Design

- The following information is required:
  - Normalized relations, including volume estimates
  - Definitions of each attribute
  - Descriptions of where and when data are **used**, **entered**, **retrieved**, **deleted**, and **updated** (including frequencies)
  - Expectations or requirements for response time and data integrity
  - Descriptions of the technologies used for implementing the files and database

Designing Fields

- **Field**: the smallest unit of named application data recognized by system software
  - Attributes from relations will be represented as fields
- **Data Type**: a coding scheme recognized by system software for representing organizational data.
  - Selecting a data type balances four objectives:
    - Minimize storage space.
    - Represent all possible values of the field.
    - Improve data integrity of the field.
    - Support all data manipulations desired on the field

TABLE 9-2 Commonly Used Data Types in Oracle 10i

Data Type	Description
VARCHAR2	Variable-length character data with a maximum length of 4000 characters; you must enter a maximum field length (e.g., VARCHAR2(30) for a field with a maximum length of 30 characters). A value less than 30 characters will consume only the required space.
CHAR	Fixed-length character data with a maximum length of 255 characters; default length is 1 character (e.g., CHAR(5) for a field with a fixed length of five characters, capable of holding a value from 0 to 5 characters long).
LONG	Capable of storing up to two gigabytes of one variable-length character data field (e.g., to hold a medical instruction or a customer comment).
NUMBER	Positive and negative numbers in the range 10^{-130} to 10^{126}; can specify the precision (total number of digits to the left and right of the decimal point) and the scale (the number of digits to the right of the decimal point) (e.g., NUMBER(5) specifies an integer field with a maximum of 5 digits and NUMBER(5, 2) specifies a field with no more than five digits and exactly two digits to the right of the decimal point).
DATE	Any date from January 1, 4712 B.C. to December 31, 4712 A.D.; date stores the century, year, month, day, hour, minute, and second.
BLOB	Binary large object, capable of storing up to four gigabytes of binary data (e.g., a photograph or sound clip).

Controlling Data Integrity

- **Default Value**: a value a field will assume unless an explicit value is entered for that field
- **Range Control**: limits range of values that can be entered into field
  - Both numeric and alphanumeric data
- **Referential Integrity**: an integrity constraint specifying that the value (or existence) of an attribute in one relation depends on the value (or existence) of the same attribute in another relation
- **Null Value**: a special field value, distinct from zero, blank, or any other value, that indicates that the value for the field is missing or otherwise unknown

File Organizations

- **File organization**: a technique for physically arranging the records of a file.
- **Physical file**: a named set of table rows stored in a contiguous section of secondary memory.

TABLE 9-3 Comparative Features of Sequential, Indexed, and Hashed File Organizations

	File Organization		
Factor	**Sequential**	**Indexed**	**Hashed**
Storage space	No wasted space	No wasted space for data, but extra space for index	Extra space may be needed to allow for addition and deletion of records
Sequential retrieval on primary key	Very fast	Moderately fast	Impractical
Random retrieval on primary key	Impractical	Moderately fast	Very fast
Multiple key retrieval	Possible, but requires scanning whole file	Very fast with multiple indexes	Not possible
Deleting rows	Can create wasted space or require reorganizing	If space can be dynamically allocated, this is easy, but requires maintenance of indexes	Very easy
Adding rows	Requires rewriting file	If space can be dynamically allocated, this is easy, but requires maintenance of indexes	Very easy, except multiple keys with same address require extra work
Updating rows	Usually requires rewriting file	Easy, but requires maintenance of indexes	Very easy

File Organizations (Cont.)

- **Sequential file organization**: a file organization in which rows in a file are stored in sequence according to a primary key value
- **Hashed file organization**: a file organization in which the address for each row is determined using an algorithm
- **Pointer:** a field of data that can be used to locate a related field or row of data

Designing Forms and Reports

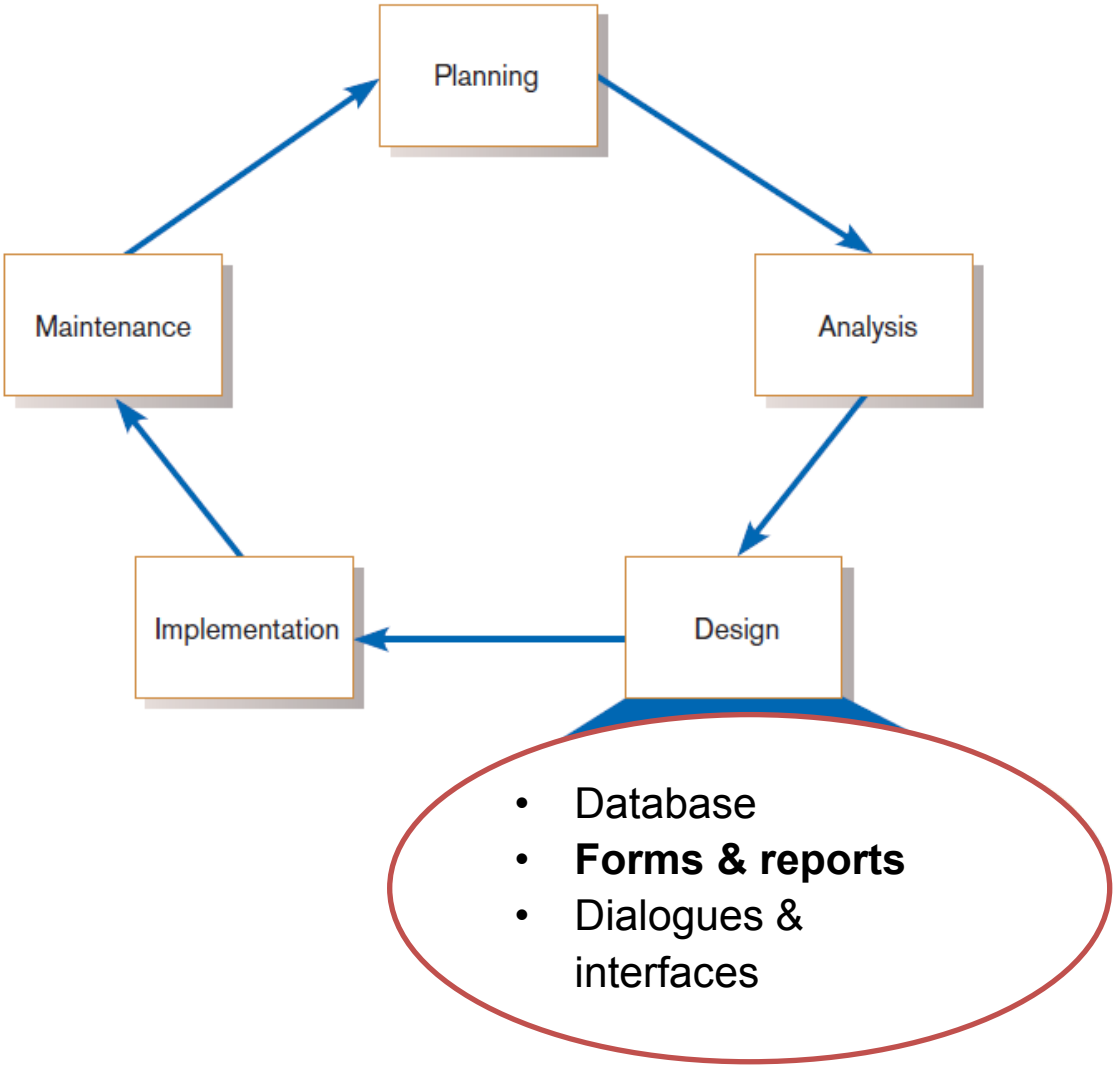


FIGURE 10-1
Systems development life cycle with logical design phase highlighted

Designing Forms and Reports (Cont.)

- **Form**: a business document that contains some predefined data and may include some areas where additional data are to be **filled** in
  - An instance of a form is typically based on one database record.
- **Report**: a business document that contains only predefined data
  - It is a passive document used solely for reading or viewing data.
- A report typically contains data from many unrelated records or transactions.

Designing Forms and Reports (Cont.)

- Common Types of Reports:
  - **Scheduled***:* produced at predefined time intervals for routine information needs
  - **Key-indicator**: provides summary of critical information on regular basis
  - **Exception**: highlights data outside of normal operating ranges
  - **Drill-down**: provides details behind summary of key-indicator or exception reports
  - **Ad-hoc**: responds to unplanned requests for non-routine information needs

The Process of Designing Forms and Reports

- Is a **user-focused** activity.
- Follows a prototyping approach.
  - First steps are to gain an understanding of the intended user and task objectives by collecting initial requirements during requirements determination.
- Requirements determination:
  - **Who** will use the form or report?
  - **What** is the purpose of the form or report?
  - **When** is the report needed or used?
  - **Where** does the form or report need to be delivered and used?
  - **How many** people need to use or view the form or report?

The Process of Designing Forms and Reports (Cont.)

- Prototyping
  - Initial prototype is designed from requirements.
  - Users review prototype design and either accept the design or request changes.
  - If changes are requested, the construction-evaluation-refinement cycle is repeated until the design is accepted.
- A coding sheet is an “old” tool for designing forms and reports, usually associated with text-based forms and reports for mainframe applications.
- Visual Basic and other development tools provide computer-aided GUI form and report generation.

The Process of Designing Forms and Reports (Cont.)

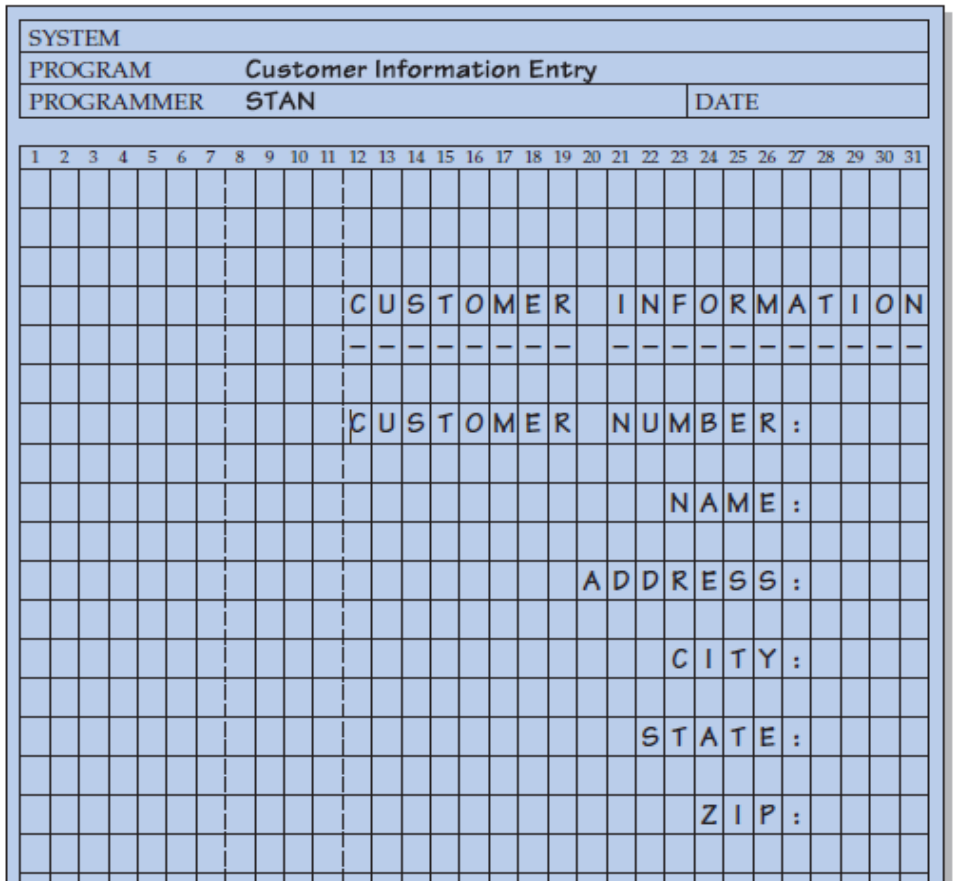
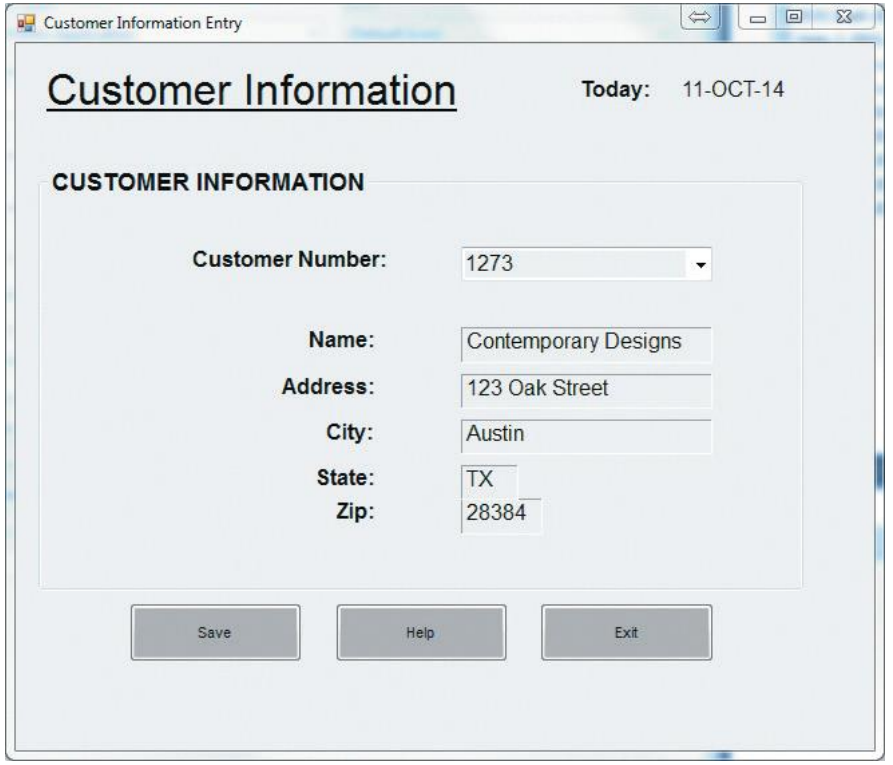


FIGURE 10-2
The layout of a data input form using a coding sheet

FIGURE 10-3
A data input screen designed in Microsoft's Visual Basic .NET (*Source:* Microsoft Corporation.)



Formatting Forms and Reports

- **Meaningful titles** — use clear, specific, version information, and current date.
- **Meaningful information** — include only necessary information, with no need to modify.
- **Balanced layout** — use adequate spacing, margins, and clear labels.
- **Easy navigation system** — show how to move forward and backward, and where you are currently

Formatting Forms and Reports (Cont.)

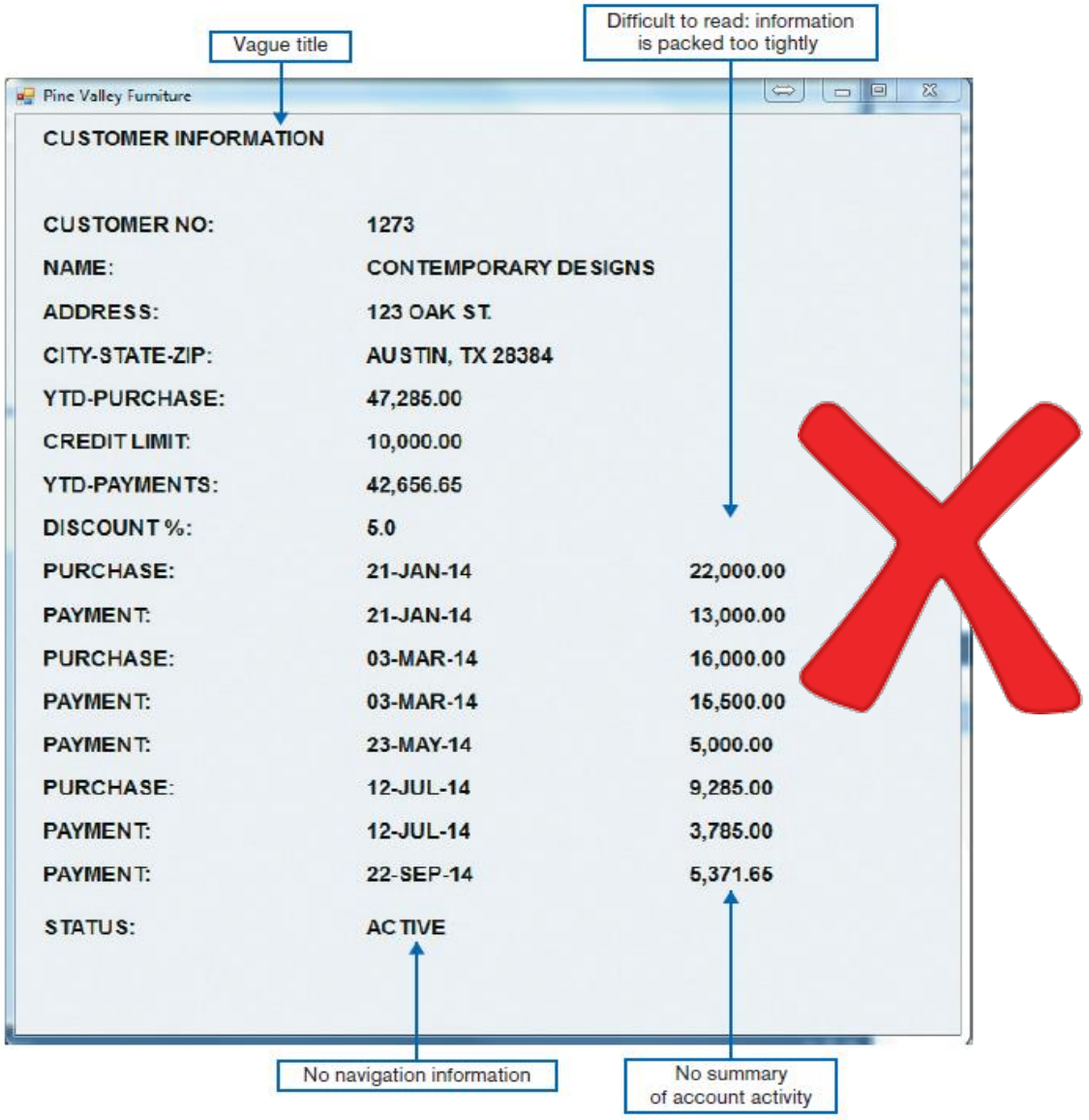


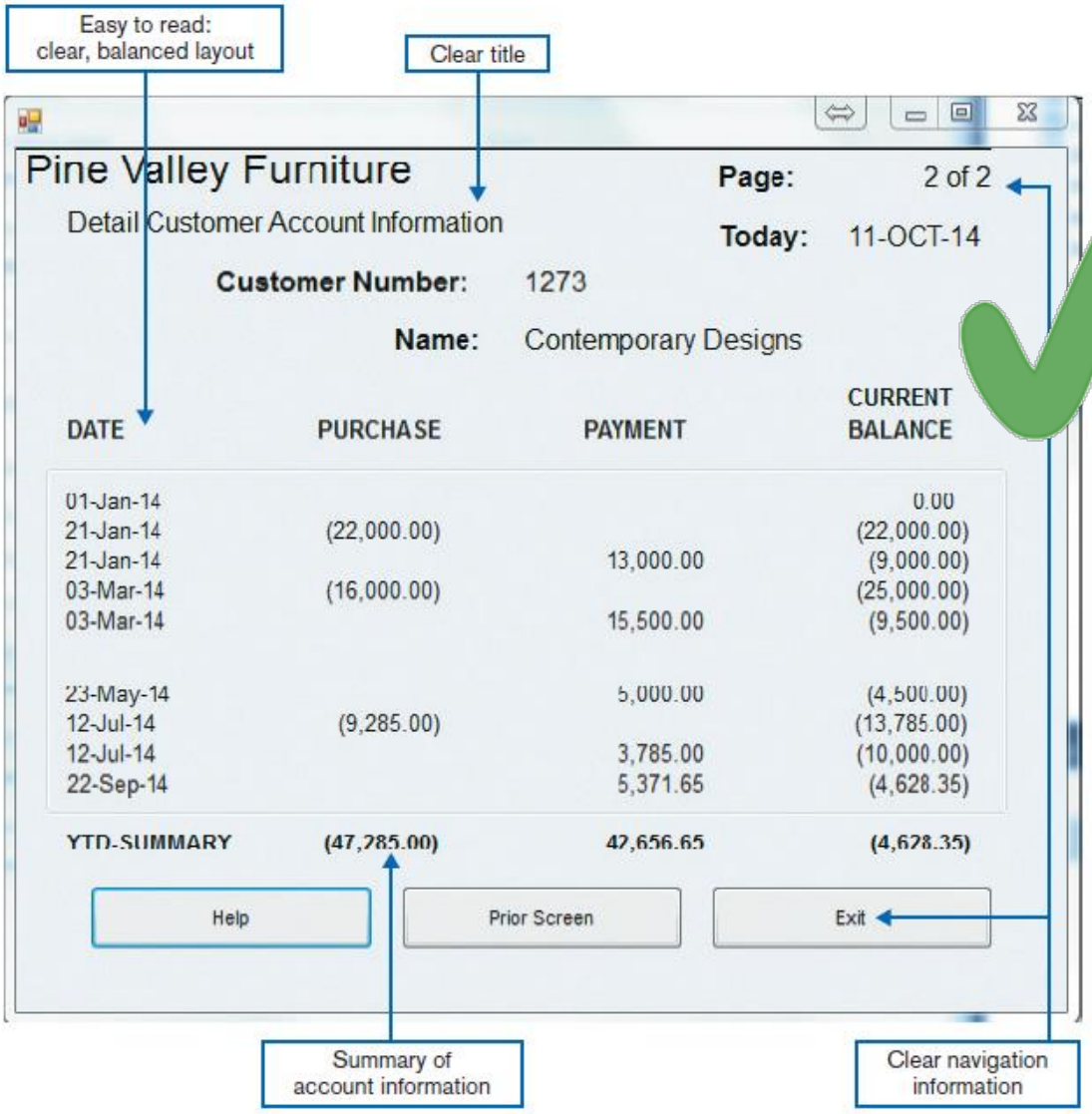
FIGURE 10-5
Contrasting customer information forms (Pine Valley Furniture) (*Source:* Microsoft Corporation.)

(a) Poorly designed form

Formatting Forms and Reports (Cont.)

FIGURE 10-5 (continued)

(b) Improved design for form



Highlighting Information

- **Notify** users of errors in data entry or processing.
- Provide **warnings** regarding possible problems.
- Draw **attention** to keywords, commands, high-priority messages, unusual data values.
- Highlighting can include use of
  - upper case
  - bold
  - italics
  - underlining
  - boxing
  - size and color differences
  - all capital letters
  - blinking
  - reverse video
  - audible tones
  - intensity differences
  - offsetting nonstandard information

Highlighting Information (Cont.)

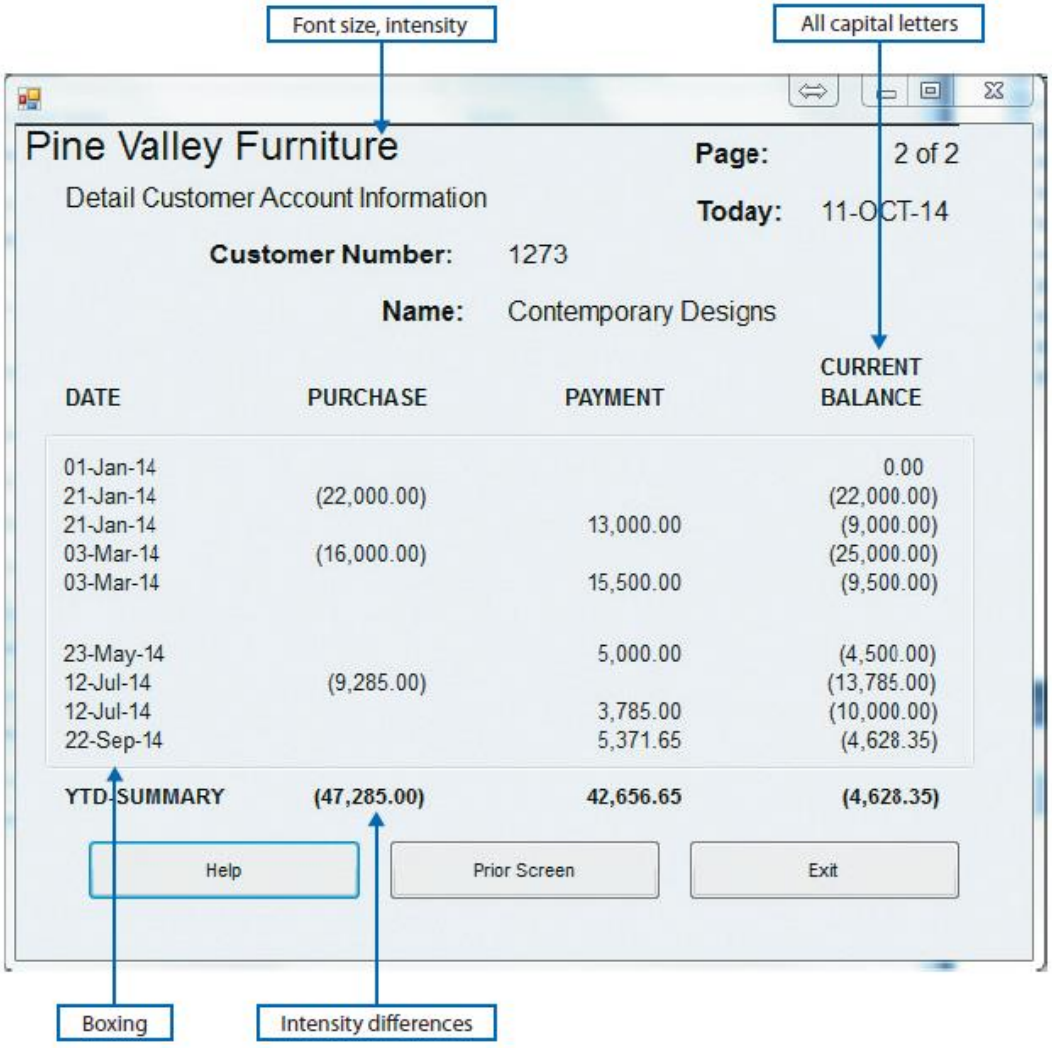


FIGURE 10-6
Customer account status display using various highlighting techniques
(Pine Valley Furniture)

(*Source:* Microsoft Corporation.)

Color vs. No Color

- Benefits — Color:
  - Soothes or strikes the eye.
  - Accents an uninteresting display.
  - Facilitates subtle discriminations in complex displays.
  - Emphasizes the logical organization of information.
  - Draws attention to warnings.
  - Evokes more emotional reactions.
- Problems from Using Color
  - Color pairings may wash out or cause problems for some users.
  - Resolution may degrade with different displays.
  - Color fidelity may degrade on different displays.
  - Printing or conversion to other media may not easily translate.

Displaying Text

- **Case**: display in mixed upper and lower case, use conventional punctuation
- **Spacing**: use double spacing if possible, otherwise blank lines between paragraphs
- **Justification**: left justify text, ragged right margins
- **Hyphenation:** don't hyphenate words between lines
- **Abbreviations**: use only when widely understood and significantly shorter than full text

Displaying Text (Cont.)

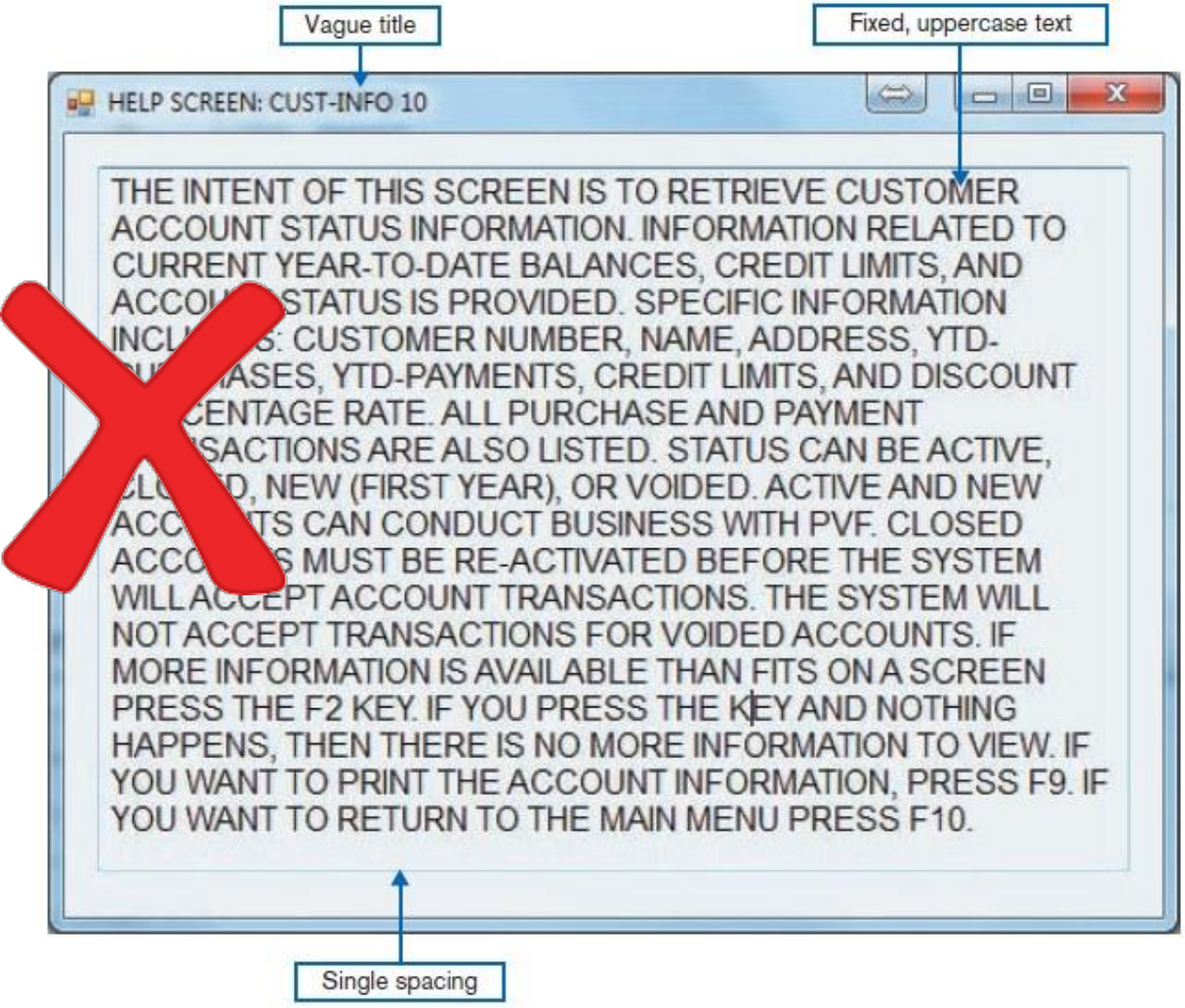


FIGURE 10-7
Contrasting the display of textual help information (*Source:* Microsoft Corporation.)

(a) Poorly designed help screen with many violations of the general guidelines for displaying text

Displaying Text (Cont.)

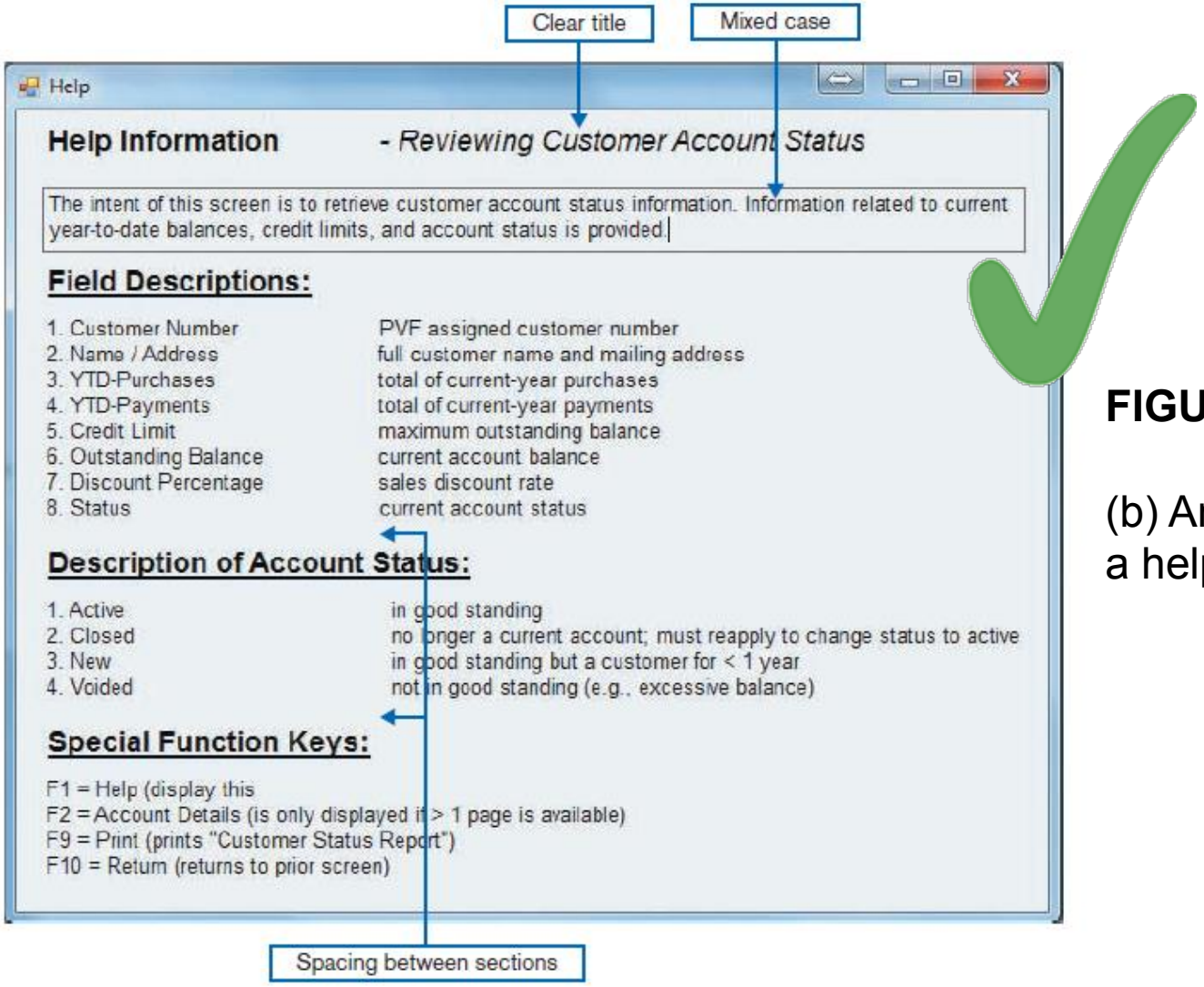


FIGURE 10-7 (continued)

(b) An improved design for a help screen

Designing Tables and Lists

- Labels
  - All columns and rows should have **meaningful** labels.
  - Labels should be separated from other information by using **highlighting**.
  - Redisplay labels when the data extend beyond a single screen or page.
- Formatting numeric, textual and alphanumeric data:
  - Right justify numeric data and **align** columns by decimal points or other delimiter.
  - Left justify textual data. Use short line length, usually 30 to 40 characters per line.
  - Break long sequences of alphanumeric data into small groups of three to four characters each.

Designing Tables and Lists (Cont.)

- Formatting columns, rows and text:
  - Sort in a meaningful **order**.
  - Place a **blank** line between every five rows in long columns.
  - **Similar** information displayed in multiple columns should be sorted vertically.
  - Columns should have at least two **spaces** between them.
  - Allow white space on printed reports for user to write notes.
  - Use a single typeface, except for emphasis.
  - Use same family of typefaces within and across displays and reports.
  - Avoid overly fancy fonts.

Designing Tables and Lists (Cont.)

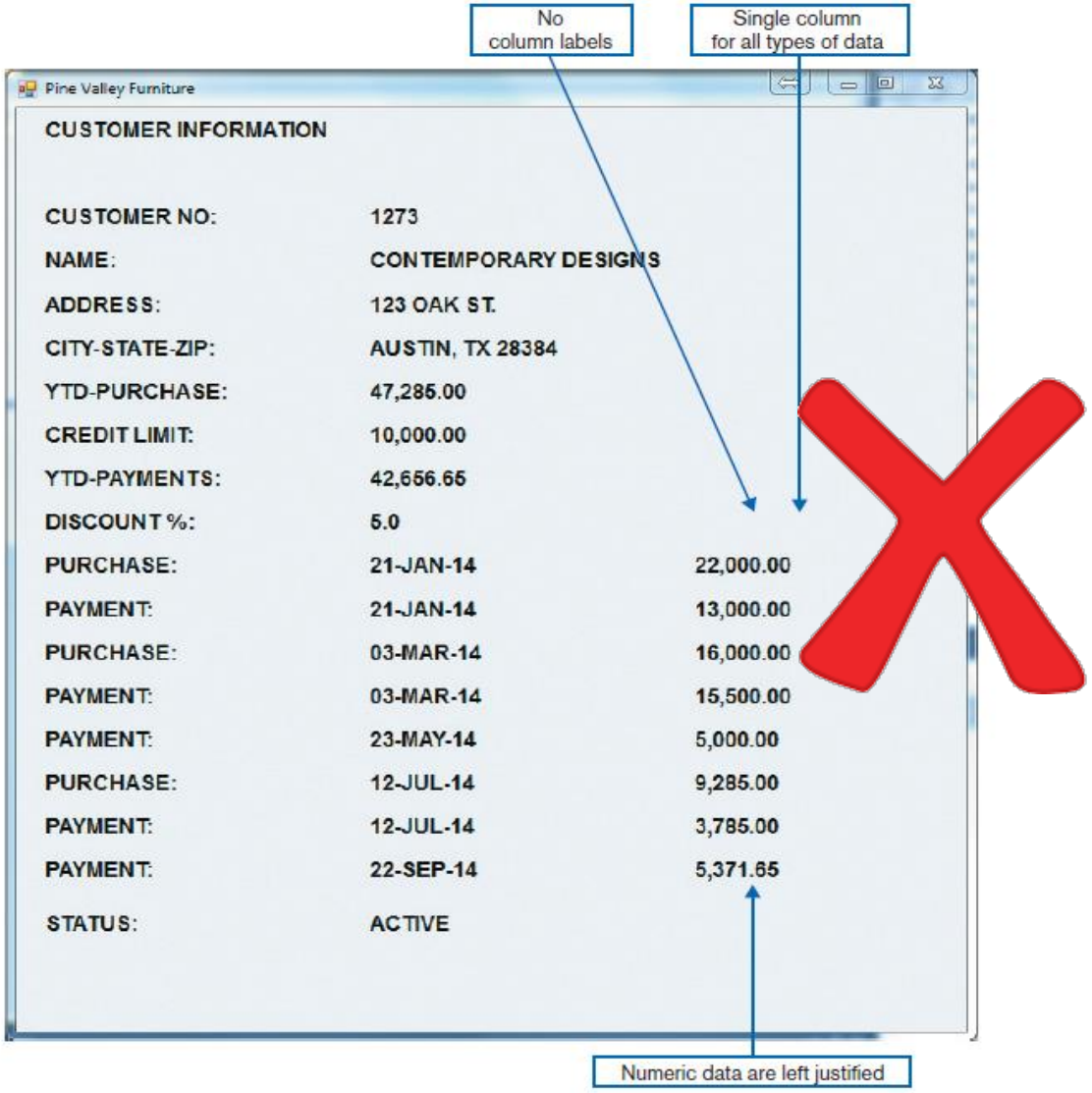


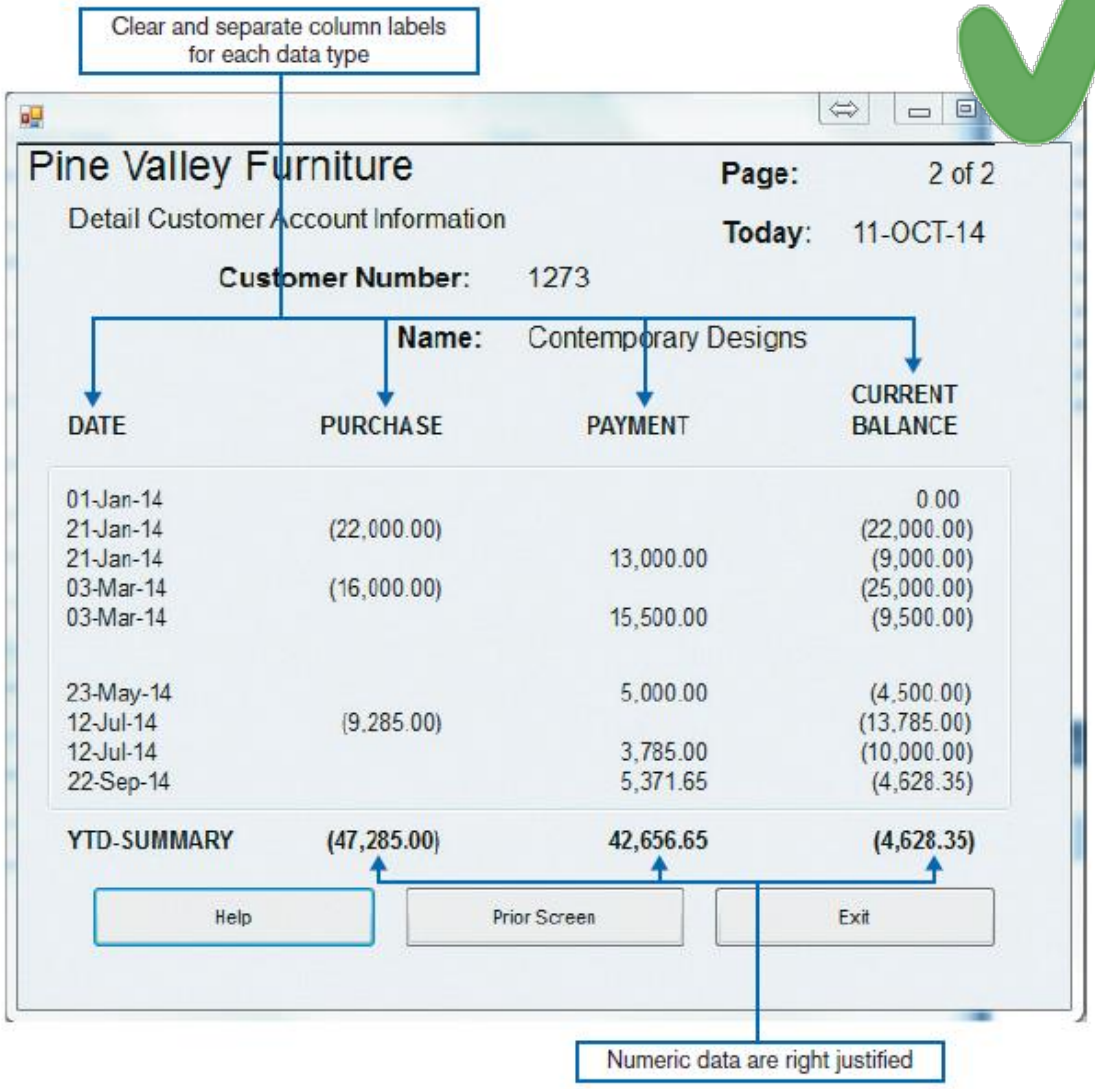
FIGURE 10-8
Contrasting the display of tables and lists (Pine Valley Furniture) (*Source:* Microsoft Corporation.)

(a) Poorly designed form

Designing Tables and Lists (Cont.)

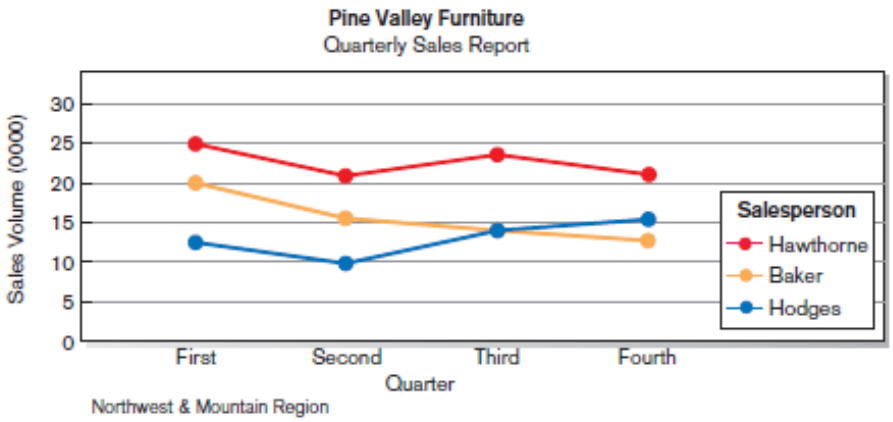
FIGURE 10-8 (continued)

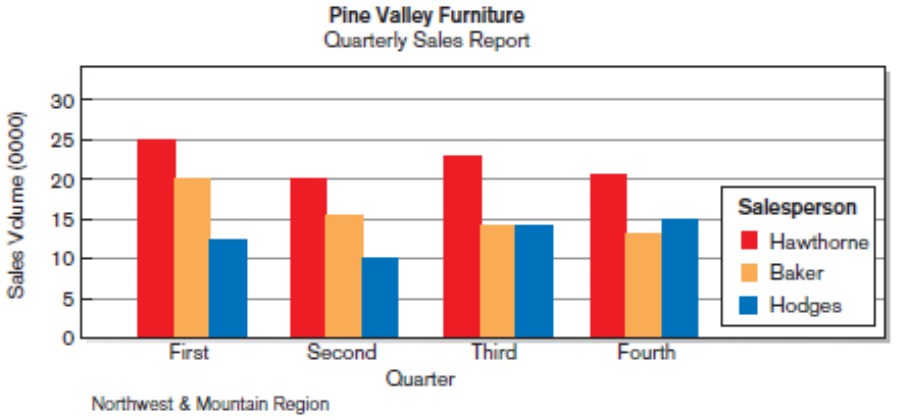
(b) Improved design for form



Designing Tables and Lists (Cont.)

- Use tables for reading individual data values.
- Use graphs for:
  - Providing **quick** summary.
  - Displaying **trends** over time.
  - Comparing points and **patterns** of variables.
  - **Forecasting** activity.
  - Simple reporting of vast quantities of information.





Designing Tables and Lists (Cont.)

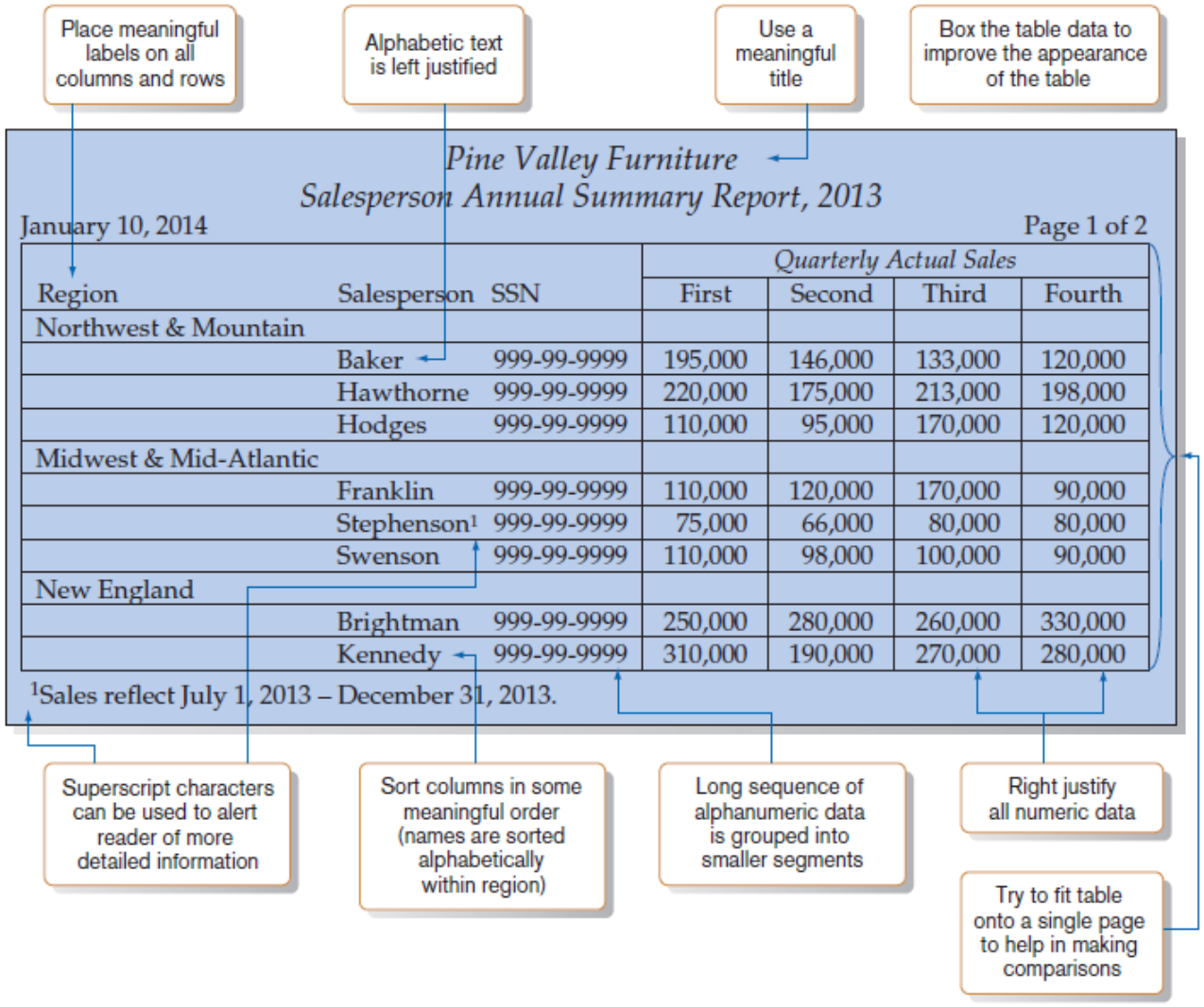


FIGURE 10-9
Tabular report illustrating numerous design guidelines

(Pine Valley Furniture)

Assessing Usability

- Objective for designing forms, reports and all human-computer interactions is **usability**.
- There are three characteristics:
  - **Speed** — Can you complete a task efficiently?
  - **Accuracy** — Does the output provide what you expect?
  - **Satisfaction** — Do you like using the output?
- **Usability**: an overall evaluation of how a system performs in supporting a particular user for a particular task.

Usability Success Factors

- **Consistency** — of terminology, formatting, titles, navigation, response time
- **Efficiency** — minimize required user actions
- **Ease** — self-explanatory outputs and labels
- **Format** — appropriate display of data and symbols
- **Flexibility** — maximize user options for data input according to preference
- Characteristics for consideration:
  - **User**: experience, skills, motivation, education, personality
  - **Task**: time pressure, cost of errors, work durations
  - **System**: platform
  - **Environment**: social and physical issues

Measures of Usability

- Time to learn
- Speed of performance
- Rate of errors
- Retention over time
- Subjective satisfaction
- Consistency of layout
  - The layout of information should be consistent both within and across applications, whether information is delivered on screen display or on a hard-copy report.

Guidelines for Usability

- **Lightweight Graphics**. the use of small, simple images to allow a Web page to more quickly be displayed
  - Quick image download
  - Quick feedback from the Web site will help to keep customers at the PVF WebStore longer
- Forms and Data Integrity Rules.
  - All forms that request information should be clearly labeled and provide adequate room for input.
  - Specific fields requiring specific information must provide a clear example.
  - Forms must designate which fields are optional, required, and which have a range of values

Designing Interfaces and Dialogues

- User-focused activity
- Prototyping methodology of iteratively:
  - Collecting information
  - Constructing a prototype
  - Assessing usability
  - Making refinements
- Must answer the who, what, where, and how questions

Designing Interfaces and Dialogues (Cont.)

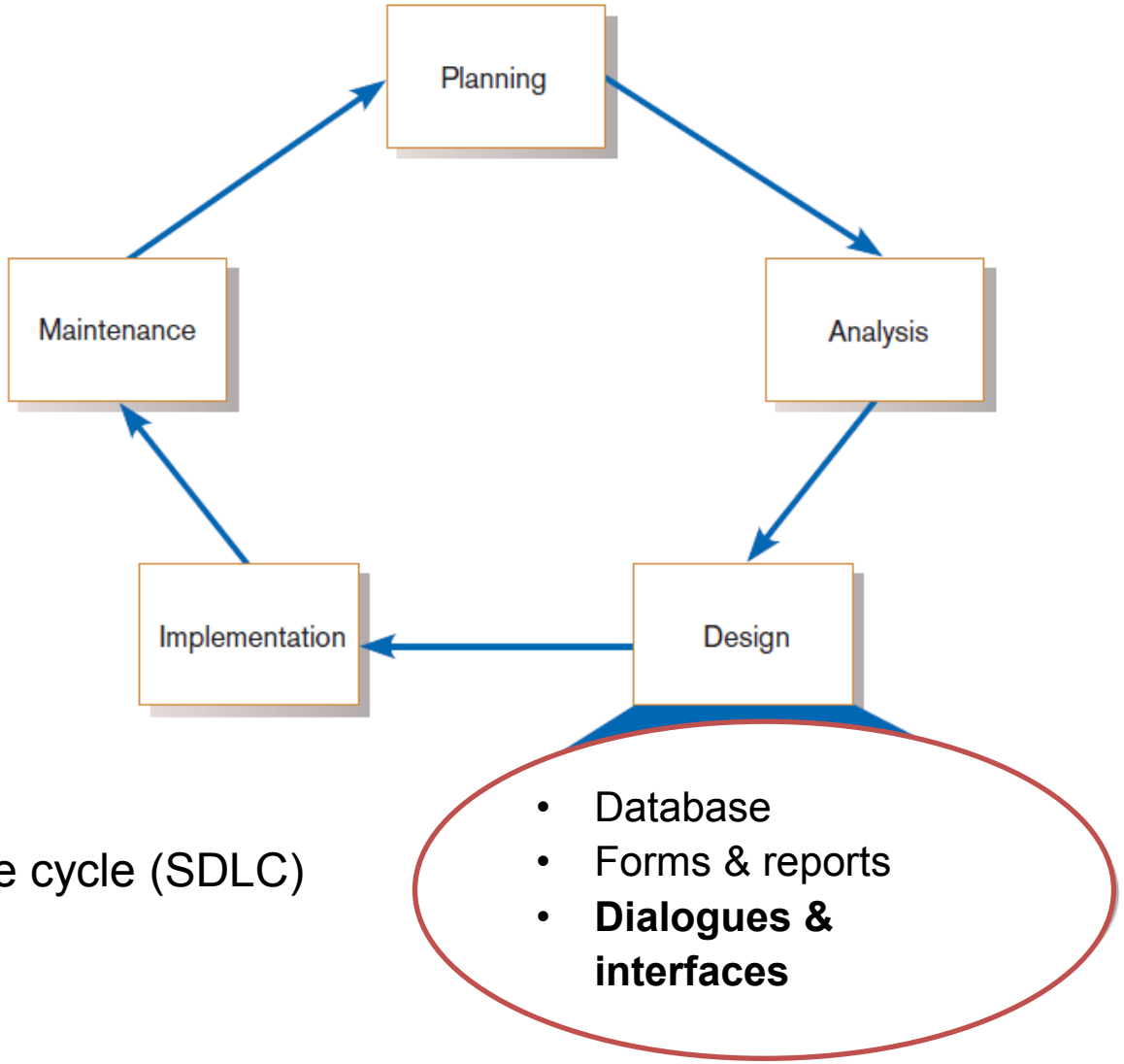


FIGURE 11-1
Systems development life cycle (SDLC)

Deliverables and Outcomes

- Creation of a design specification
  - A typical interface/dialogue design specification is similar to form design, but includes **multiple** forms and dialogue sequence specifications.
- The specification includes:
  - Narrative overview
  - Sample design
  - Testing and usability assessment
  - Dialogue sequence
- **Dialogue sequence**—the ways a user can move from one display to another

Design Specification

1. Narrative Overview
   a. Interface/Dialogue Name
   b. User Characteristics
   c. Task Characteristics
   d. System Characteristics
   e. Environmental Characteristics

2. Interface/Dialogue Designs
   a. Form/Report Designs
   b. Dialogue Sequence Diagram(s) and Narrative Description

3. Testing and Usability Assessment
   a. Testing Objectives
   b. Testing Procedures
   c. Testing Results
      i) Time to Learn
      ii) Speed of Performance
      iii) Rate of Errors
      iv) Retention over Time
      v) User Satisfaction and Other Perceptions

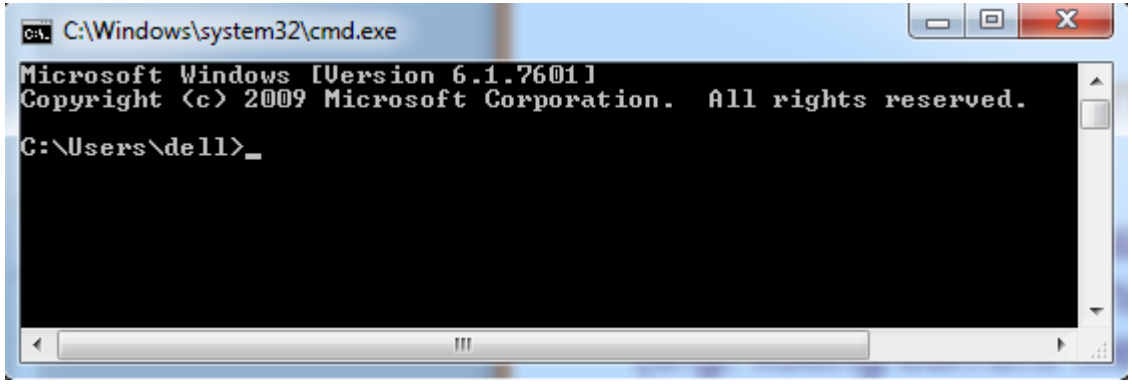
Figure 11-2
Specification outline for the design of interfaces and dialogues

Interaction Methods and Devices

- **Interface**: a method by which users **interact** with an information system
- All human-computer interfaces must:
  - have an interaction style, and
  - use some hardware device(s) for supporting this interaction.
- Methods of Interacting
  - Command line
    - Includes keyboard shortcuts and function keys
  - Menu
  - Form
  - Object-based
  - Natural language

Command Language Interaction

- **Command language interaction:** a human-computer interaction method whereby users enter explicit statements into a system to invoke operations
- Example from MS DOS:
  - COPY C:PAPER.DOC A:PAPER.DOC
  - Command copies a file from C: drive to A: drive
- Large overhead to **learning** set of commands



Menu Interaction

- **Menu interaction**: a human-computer interaction method in which a list of system options is provided and a specific command is invoked by user selection of a menu option
- **Pop-up menu**: a menu-positioning method that places a menu near the current cursor position.
- **Drop-down menu** is a menu-positioning method that places the access point of the menu near the top line of the display.
  - When accessed, menus open by dropping down onto the display.
  - Visual editing tools help designers construct menus.

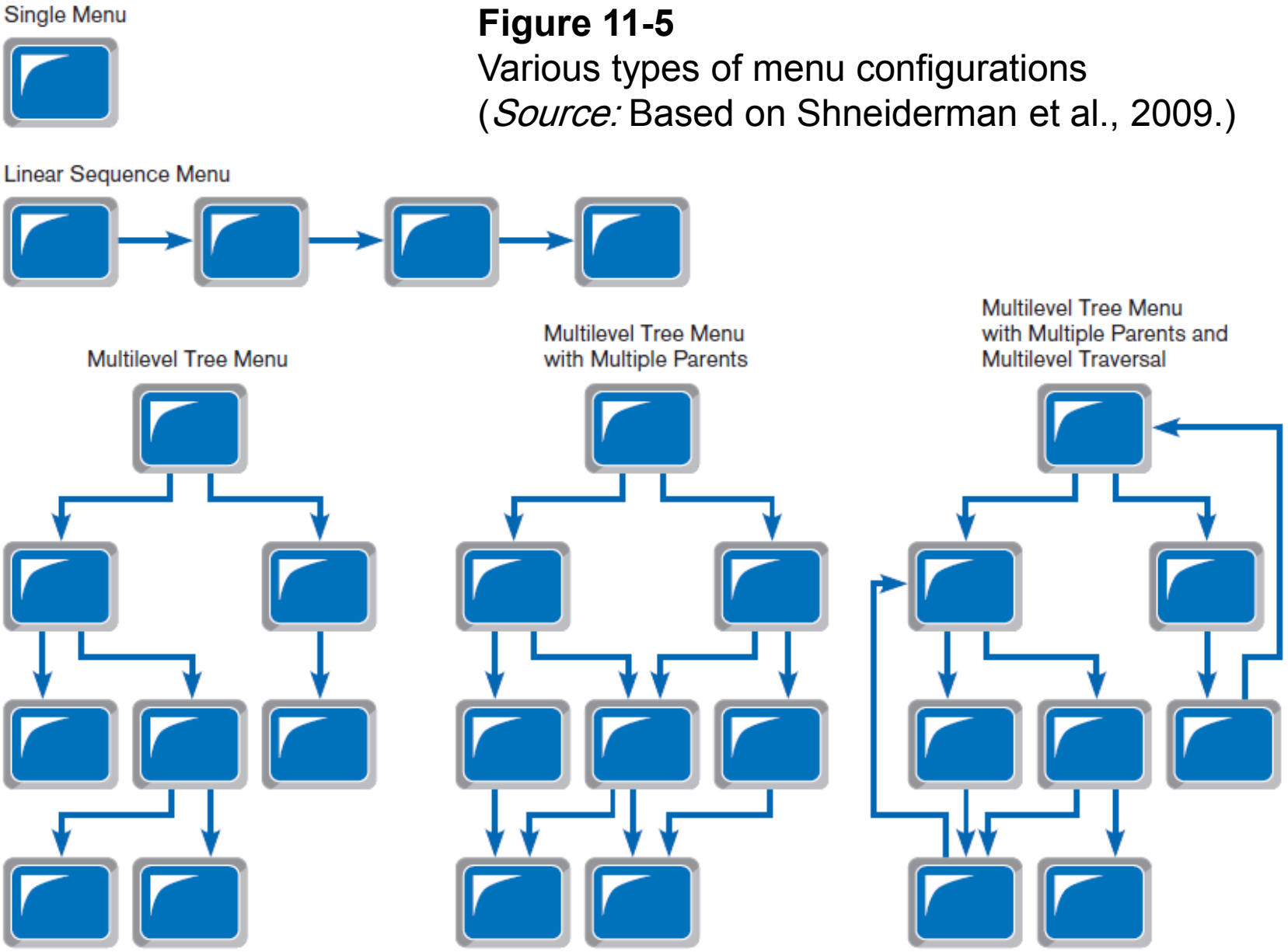


Figure 11-5
Various types of menu configurations
(*Source:* Based on Shneiderman et al., 2009.)

Menu Interaction (Cont.)

- Guidelines for Menu Design
  - **Wording** — meaningful titles, clear command verbs, mixed upper/lower case
  - **Organization** — consistent organizing principle
  - **Length** — all choices fit within screen length
  - **Selection** — consistent, clear and easy selection methods
  - **Highlighting** — only for selected options or unavailable options

Menu Interaction (Cont.)

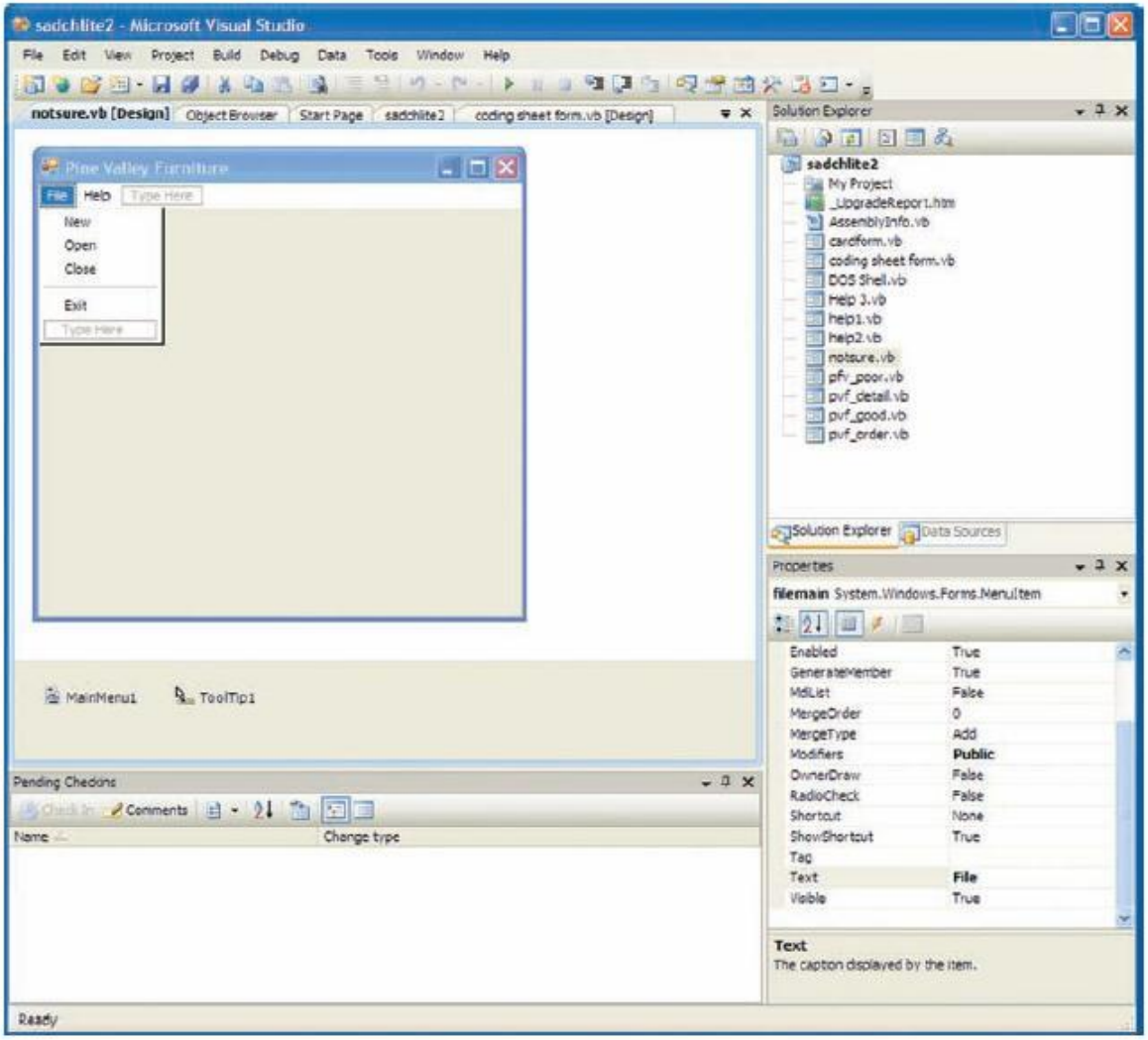
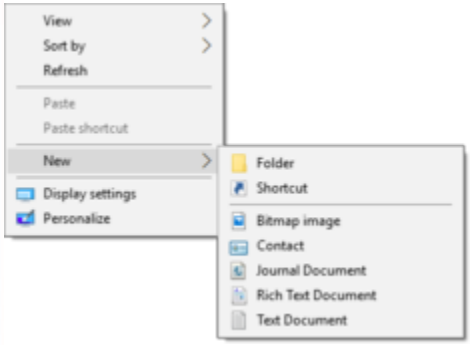
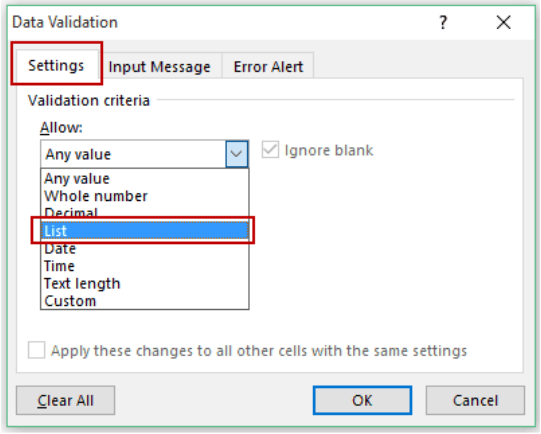


FIGURE 11-8
Menu building with Microsoft Visual Basic .NET



Pop-up menu



Drop-down menu

Form Interaction

- **Form interaction**: a highly intuitive human-computer interaction method whereby **data fields** are formatted in a manner similar to paper-based forms
  - Allows users to fill in the blanks when working with a system.

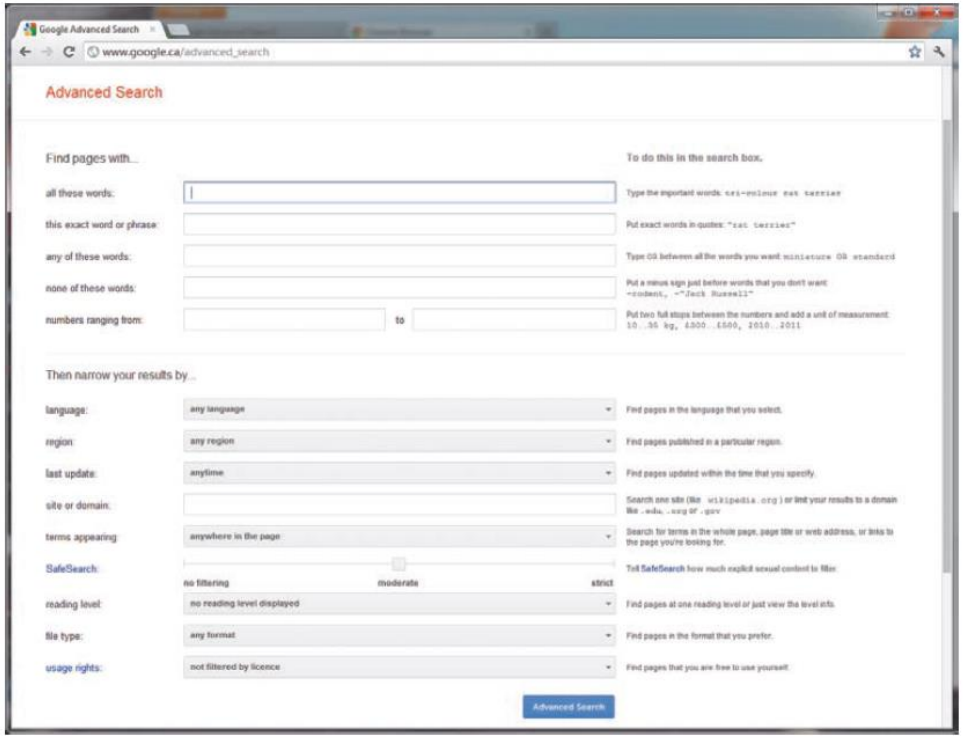


FIGURE 11-9
Example of form interaction from the Google Advanced Search Engine (*Source: Google.)*

Object-Based Interaction

- **Object-based interaction**: a human-computer interaction method in which **symbols** are used to represent commands or functions
- **Icons**: graphical pictures that represent specific functions within a system
  - Use little screen space and are easily understood by users

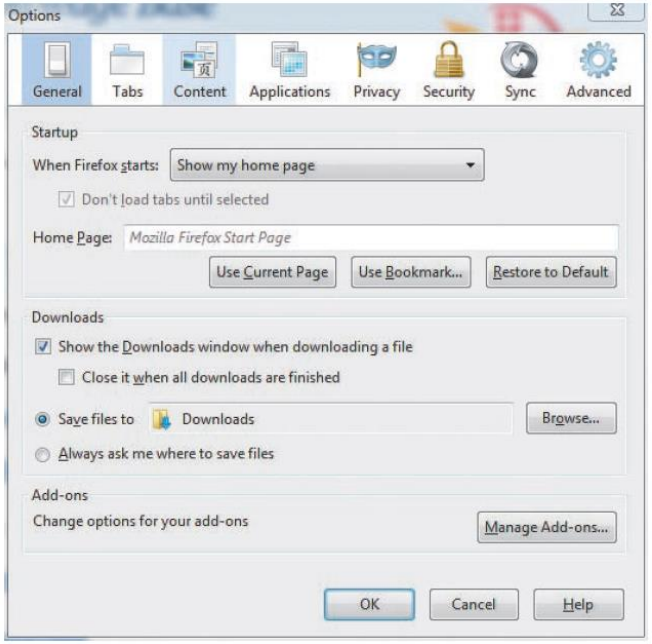


Figure 11-10
Object-based (icon) interface from the Option menu in the Firefox Web browser

Natural Language Interaction

- **Natural language interaction**: a human-computer interaction method whereby inputs to and outputs from a computer-based application are in a conventional **spoken language** such as English
- Based on research in artificial intelligence
- Current implementations are tedious and difficult to work with, not as viable as other interaction methods.

Hardware Options for System Interaction

- Keyboard
- Mouse
- Joystick
- Trackball
- Touch screen
- Light Pen
- Graphics Tablet
- Voice

Usability Problems with Hardware Devices

- **Visual Blocking**. Extent to which device blocks display when using
- **User Fatigue**. Potential for fatigue over long use
- **Movement Scaling**. Extent to which device movement translates to equivalent screen movement
- **Durability**. Lack of durability or need for maintenance (e.g., cleaning) over extended use
- **Adequate Feedback**. Extent to which device provides adequate feedback for each operation
- **Speed**. Cursor movement speed
- **Pointing Accuracy**. Ability to precisely direct cursor

Usability Problems with Hardware Devices (Cont.)

TABLE 11-3 Summary of Interaction Device Usability Problems

	Problem						
Device	Visual Blocking	User Fatigue	Movement Scaling	Durability	Adequate Feedback	Speed	Pointing Accuracy
Keyboard	□	□	■	□	■	■	□
Mouse	□	□	■	□	■	□	□
Joystick	□	□	■	□	■	□	■
Trackball	□	□	■	■	■	□	□
Touch Screen	■	■	□	■	□	□	■
Light Pen	■	■	□	□	□	□	■
Graphics Tablet	□	□	■	□	■	□	□
Voice	□	□	■	□	■	□	■

Key:

□ = little or no usability problems

■ = potentially high usability problems for some applications

Usability Problems with Hardware Devices (Cont.)

TABLE 11-4 Summary of General Conclusions from Experimental Comparisons of Input Devices in Relation to Specific Task Activities

Task	Most Accurate	Shortest Positioning	Most Preferred
Target Selection	trackball, graphics tablet, mouse, joystick	touch screen, light pen, mouse, graphics tablet, trackball	touch screen, light pen
Text Selection	mouse	mouse	—
Data Entry	light pen	light pen	—
Cursor Positioning	—	light pen	—
Text Correction	light pen, cursor keys	light pen	light pen
Menu Selection	touch screen	—	keyboard, touch screen

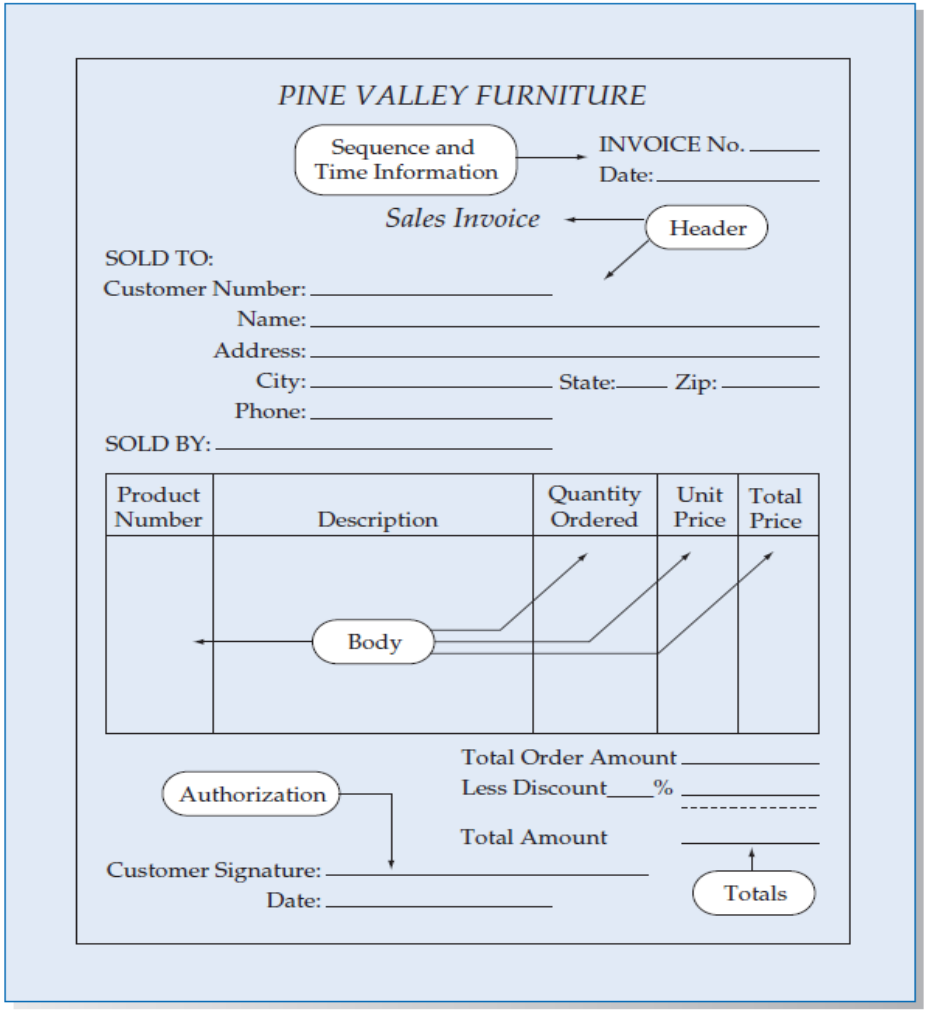
Key:

Target Selection = moving the cursor to select a figure or item
Text Selection = moving the cursor to select a block of text
Data Entry = entering information of any type into a system
Cursor Positioning = moving the cursor to a specific position
Text Correction = moving the cursor to a location to make a text correction
Menu Selection = activating a menu item
— = no clear conclusion from the research

(*Source*: Based on Blattner and Schultz, 1988.)

Designing Interfaces

- Forms have several general areas in common:
  - **Header** information
  - **Sequence** and time-related information
  - Instruction or **formatting** information
  - **Body** or data details
  - **Totals** or data summary
  - Authorization or **signatures**
  - Comments



Paper-based form for reporting customer sales activity (Pine Valley Furniture)

Designing Interfaces (Cont.)

- Use standard formats similar to paper-based forms and reports.
- Use left-to-right (or right-to-left for Persain), top-to-bottom navigation.
- Flexibility and consistency:
  - Free movement between fields
  - No permanent data storage until the user requests
  - Each key and command assigned to one function

Structuring Data Entry

Entry	Never require data that are already online or that can be computed
Defaults	Always provide default values when appropriate
Units	Make clear the type of data units requested for entry
Replacement	Use character replacement when appropriate
Captioning	Always place a caption adjacent to fields
Format	Provide formatting examples
Justify	Automatically justify data entries
Help	Provide context-sensitive help when appropriate

Controlling Data Input

- Objective: Reduce data entry **errors**
- Common sources of data entry errors in a field:
  - Appending: adding additional characters
  - Truncating: losing characters
  - Transcripting: entering invalid data
  - Transposing: reversing sequence of characters

TABLE 11-9 Validation Tests and Techniques to Enhance the Validity of Data Input

Validation Test	Description
Class or Composition	Test to ensure that data are of proper type (e.g., all numeric, all alphabetic, all alphanumeric)
Combinations	Test to see if the value combinations of two or more data fields are appropriate or make sense (e.g., does the quantity sold make sense given the type of product?)
Expected Values	Test to see if data are what is expected (e.g., match with existing customer names, payment amount, etc.)
Missing Data	Test for existence of data items in all fields of a record (e.g., is there a quantity field on each line item of a customer order?)
Pictures/Templates	Test to ensure that data conform to a standard format (e.g., are hyphens in the right places for a student ID number?)
Range	Test to ensure data are within proper range of values (e.g., is a student's grade point average between 0 and 4.0?)
Reasonableness	Test to ensure data are reasonable for situation (e.g., pay rate for a specific type of employee)
Self-Checking Digits	Test where an extra digit is added to a numeric field in which its value is derived using a standard formula (see Figure 11-14)
Size	Test for too few or too many characters (e.g., is social security number exactly nine digits?)
Values	Test to make sure values come from set of standard values (e.g., two-letter state codes)

Providing Feedback

- Three types of system feedback:
  - **Status information**: keep user informed of what's going on, helpful when user has to wait for response.
  - **Prompting cues**: tell user when input is needed, and how to provide the input.
  - **Error or warning messages**: inform user that something is wrong, either with data entry or system operation

Providing Help

- Place yourself in user's place when designing help.
- Guidelines for designing usable help:
  - **Simplicity** — Help messages should be short and to the point.
  - **Organize** — Information in help messages should be easily absorbed by users.
  - **Show** — It is useful to explicitly show users how to perform an operation.

TABLE 11-12 Types of Help

Type of Help	Example of Question
Help on Help	How do I get help?
Help on Concepts	What is a customer record?
Help on Procedures	How do I update a record?
Help on Messages	What does "Invalid File Name" mean?
Help on Menus	What does "Graphics" mean?
Help on Function Keys	What does each Function key do?
Help on Commands	How do I use the "Cut" and "Paste" commands?
Help on Words	What do "Merge" and "Sort" mean?

Designing Dialogues

- **Dialogue**: the sequence of interaction between a user and a system
- Dialogue design involves:
  - Designing a dialogue sequence.
  - Building a prototype.
  - Assessing usability.
- Designing the Dialogue Sequence
  - Typical dialogue between user and Customer Information System:
    - Request to view individual customer information.
    - Specify the customer of interest.
    - Select the year-to-date transaction summary display.
    - Review the customer information.
    - Leave system.

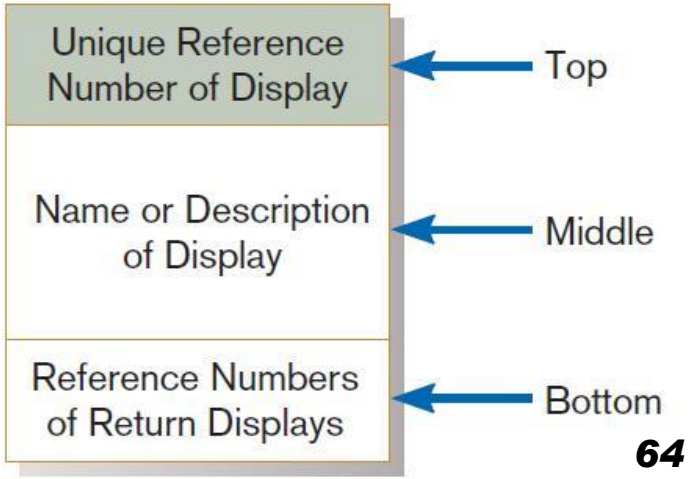
Guidelines for Designing Human-Computer Dialogues

- Consistency
- Shortcuts and Sequence
- Feedback
- Closure
- Error Handling
- Reversal
- Control
- Ease

Designing the Dialogue Sequence (Cont.)

- **Dialogue diagramming**: a formal method for designing and representing human-computer dialogues using box and line diagrams.
- Three sections of the box:
  - **Top**—contains a unique display reference number used by other displays for referencing it
  - **Middle**—contains the name or description of the display
  - **Bottom**—contains display reference numbers that can be accessed from the current display

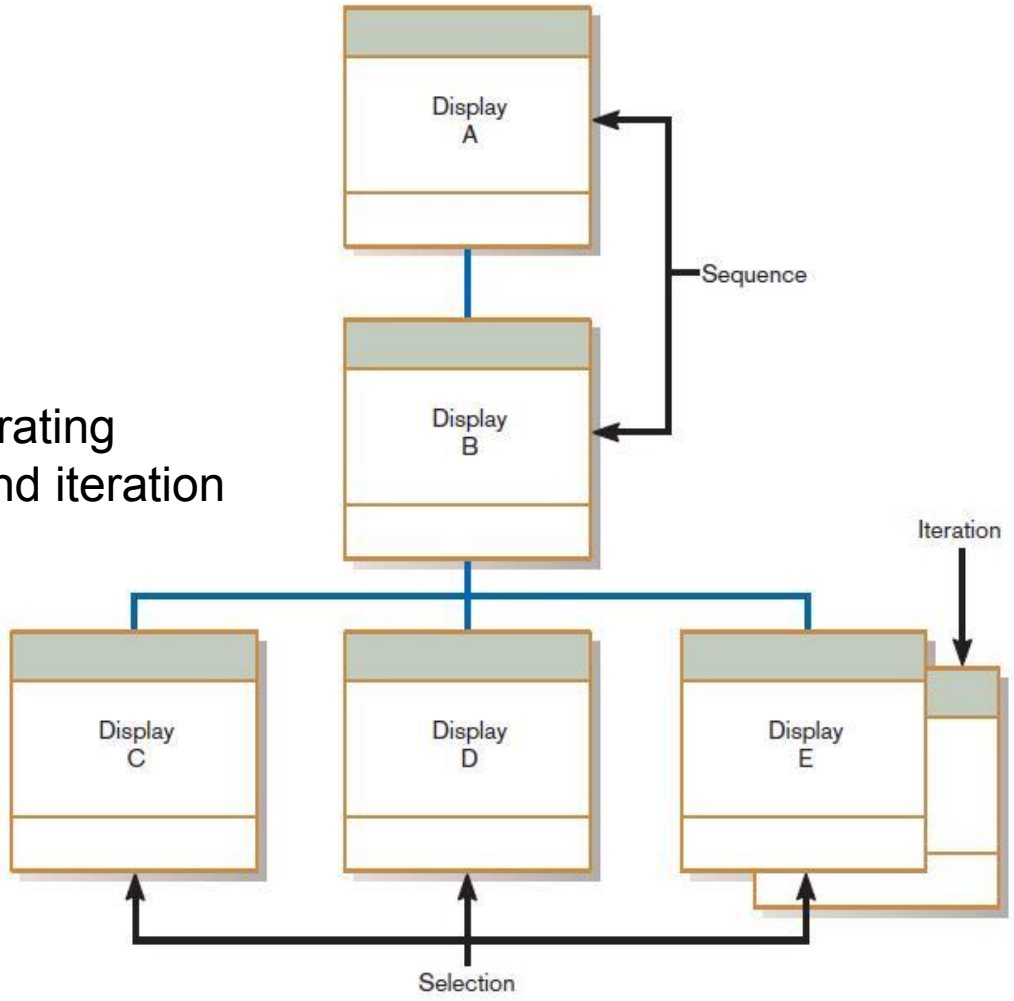
Sections of a dialogue diagramming box



- Dialogue diagrams depict the sequence, conditional branching, and repetition of dialogues.

Designing the Dialogue Sequence (Cont.)

FIGURE 11-18
Dialogue diagram illustrating sequence, selection, and iteration



Graphical Interface Design Issues

- Become an expert user of the GUI environment.
  - Understand how other applications have been designed.
  - Understand standards.
- Understand the available resources and how they can be used.
  - Become familiar with standards for menus and forms.

Graphical Interface Design Issues (Cont.)

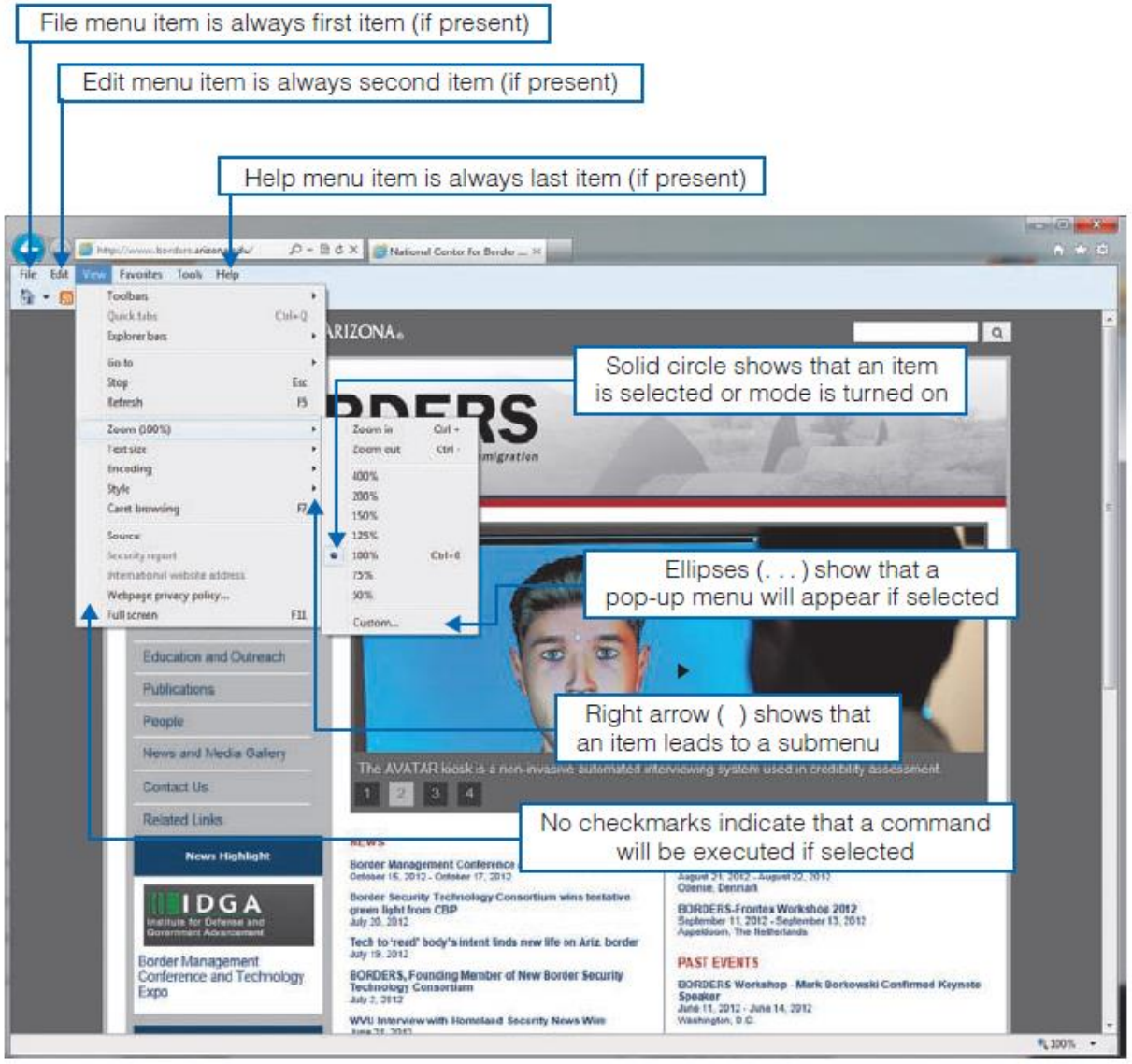


Figure 11-20
Highlighting GUI design standards (*Source:* University of Arizona.)

General Guidelines

- Web's single "click-to-act" method of loading static hypertext documents (i.e. most buttons on the Web do not provide click feedback)
- Limited capabilities of most Web browsers to support finely grained user interactivity.
- Limited agreed-upon standards for encoding Web content and control mechanisms
- Lack of maturity of Web scripting and programming languages as well as limitations in commonly used Web GUI component libraries
- **Cookie crumbs**: the technique of placing "tabs" on a Web page that show a user where he or she is on a site and where he or she has been
  - Allow users to navigate to a point previously visited and will assure they are not lost
  - Clearly show users where they have been and how far they have gone from home

Common Errors in Web site Design

- Opening new browser window
- Breaking or slowing down the Back button
- Complex URLs
- Orphan Pages
- Scrolling navigation pages
- Lack of navigation support
- Hidden links
- Links that don't provide enough information
- Buttons that provide no click feedback

Summary

- Explain the role of designing databases in the analysis and design of an information system.
- Transform an entity-relationship (E-R) diagram into an equivalent set of well-structured (normalized) relations.
- Choose storage formats for fields in database tables.
- Translate well-structured relations into efficient database tables.
- Explain when to use different types of file organizations to store computer files.
- Explain the process of designing forms and reports and the deliverables for their creation.
- Apply the general guidelines for formatting forms and reports.
- Explain how to assess usability and describe how variations in users, tasks, technology, and environmental characteristics influence the usability of forms and reports.
- Explain the process of designing interfaces and dialogues and the deliverables for their creation.
- Describe and apply the general guidelines for designing interfaces and specific guidelines for layout design, structuring data entry fields, providing feedback, and system help.

References

- **Chapter 9-11,** J. S. Valacich, J. George, **Modern Systems Analysis and Design**. 8th Edition, Pearson 2017.