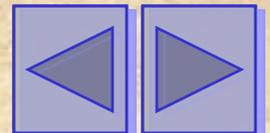
The background is a light beige, textured surface. A blue horizontal line spans the width of the page, with a small blue circle at its left end. A blue vertical line descends from the top of this circle. Another blue horizontal line is positioned below the main title, with a small blue circle at its right end. A blue vertical line descends from the top of this circle.

Chapter 4.1.

Designing Databases

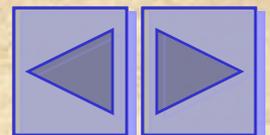
Purpose of Database Design

- ◆ Structure the data in stable structures, called normalized tables
 - Not likely to change over time
 - Minimal redundancy
- ◆ Develop a logical database design that reflects actual data requirements
- ◆ Develop a logical database design from which a physical database design can be developed



Purpose of Database Design

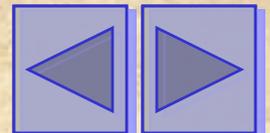
- ◆ Translate a relational database model into a technical file and database design that balances several performance factors
- ◆ Choose data storage technologies that will efficiently, accurately and securely process database activities



Process of Database Design

◆ Logical Design

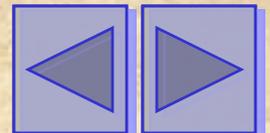
- Based upon the conceptual data model
- Four key steps
 1. Develop a logical data model for each known user interface for the application using normalization principles
 2. Combine normalized data requirements from all user interfaces into one consolidated logical database model
 3. Translate the conceptual 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



Process of Database Design

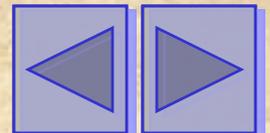
◆ Physical Design

- Based upon results of logical database design
- Key decisions
 1. Choosing storage format for each attribute from the logical database model
 2. Grouping attributes from the logical database model into physical records
 3. Arranging related records in secondary memory (hard disks and magnetic tapes) so that records can be stored, retrieved and updated rapidly
 4. Selecting media and structures for storing data to make access more efficient



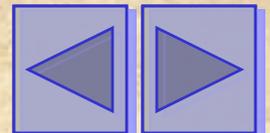
Deliverables and Outcomes

- ◆ Logical database design must account for every data element on a system input or output
- ◆ Normalized relations are the primary deliverable
- ◆ Physical database design results in converting relations into files



Relational Database Model

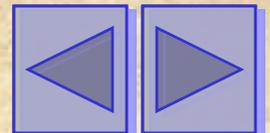
- ◆ Data represented as a set of related tables or relations
- ◆ Relation
 - A named, two-dimensional table of data. Each relation consists of a set of named columns and an arbitrary number of unnamed rows
 - Properties
 - ◆ Entries in cells are simple
 - ◆ Entries in columns are from the same set of values
 - ◆ Each row is unique
 - ◆ The sequence of columns can be interchanged without changing the meaning or use of the relation
 - ◆ The rows may be interchanged or stored in any sequence



Relational Database Model

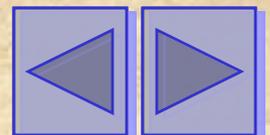
◆ Well-Structured Relation

- A relation that contains a minimum amount of redundancy and allows users to insert, modify and delete the rows without errors or inconsistencies



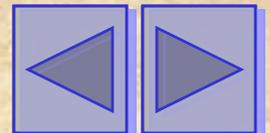
Normalization

- ◆ The process of converting complex data structures into simple, stable data structures is normalization
- ◆ First Normal Form (1NF)
 - Contains only atomic values
- ◆ Second Normal Form (2NF)
 - It is already in 1NF
 - Each nonprimary key attribute is identified by the whole key (called full functional dependency)



Normalization

- ◆ Third Normal Form (3NF)
 - It is already in 2NF
 - Nonprimary key attributes do not depend on each other (called transitive dependencies)
- ◆ The result of normalization is that every nonprimary key attribute depends upon the whole primary key



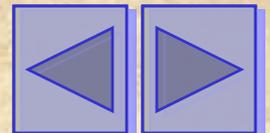
Functional Dependencies and Primary Keys

◆ Functional Dependency

- A particular relationship between two attributes. For a given relation, attribute B is functionally dependent on attribute A is, for every valid value of A, that value of A uniquely determines the value of B
- Instances (or sample data) in a relation do not prove the existence of a functional dependency
- Knowledge of problem domain is most reliable method for identifying functional dependency

◆ Primary Key

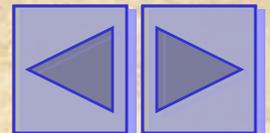
- An attribute whose value is unique across all occurrences of a relation



Functional Dependencies and Primary Keys

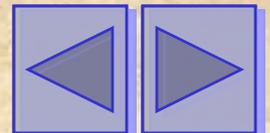
◆ Second Normal Form (2NF)

- A relation is in second normal form (2NF) if any of the following conditions apply:
 - ◆ The primary key consists of only one attribute
 - ◆ Nonprimary key attributes exist in the relation
 - ◆ Every nonprimary key attribute is functionally dependent on the full set of primary key attributes



Functional Dependencies and Primary Keys

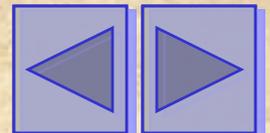
- ◆ Conversion to second normal form (2NF)
 - To convert a relation into 2NF, decompose the relation into new relations using the attributes, called determinants, that determine other attributes
 - The determinants become the primary key of the new relation



Functional Dependencies and Primary Keys

◆ Third Normal Form (3NF)

- A relation is in third normal form (3NF) if it is in second normal form (2NF) and there are no functional (transitive) dependencies between two (or more) nonprimary key attributes



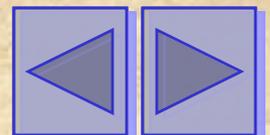
Functional Dependencies and Primary Keys

◆ Foreign Key

- An attribute that appears as a nonprimary key attribute in one relation and as a primary key attribute (or part of a primary key) in another relation

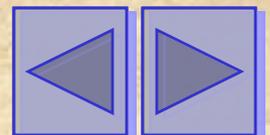
◆ Referential Integrity

- 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



Transforming E-R Diagrams into Relations

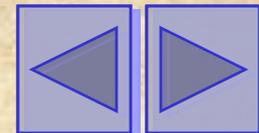
- ◆ It is useful to transform the conceptual data model into a set of normalized relations
- ◆ Steps
 - Represent entities
 - Represent relationships
 - Normalize the relations
 - Merge the relations



Transforming E-R Diagrams into Relations

◆ Represent Entities

- Each regular entity is transformed into a relation
- The identifier of the entity type becomes the primary key of the corresponding relation
- The primary key must satisfy the following two conditions
 - a. The value of the key must uniquely identify every row in the relation
 - b. The key should be nonredundant



Transforming E-R Diagrams into Relations

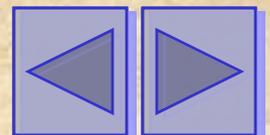
◆ Represent Relationships

■ Binary 1:N Relationships

- ◆ Add the primary key attribute (or attributes) of the entity on the one side of the relationship as a foreign key in the relation on the right side
- ◆ The one side *migrates* to the many side

■ Binary or Unary 1:1

- ◆ Three possible options
 - a. Add the primary key of A as a foreign key of B
 - b. Add the primary key of B as a foreign key of A
 - c. Both of the above



Transforming E-R Diagrams into Relations

- ◆ Represent Relationships (continued)
 - Binary and Higher M:N relationships
 - ◆ Create another relation and include primary keys of all relations as primary key of new relation
 - Unary 1:N Relationships
 - ◆ Relationship between instances of a single entity type
 - ◆ Utilize a recursive foreign key
 - A foreign key in a relation that references the primary key values of that same relation
 - Unary M:N Relationships
 - ◆ Create a separate relation
 - ◆ Primary key of new relation is a composite of two attributes that both take their values from the same primary key

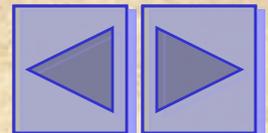
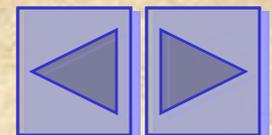


TABLE 12-1 E-R to Relational Transformation

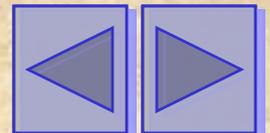
<i>E-R Structure</i>	<i>Relational Representation</i>
Regular entity	Create a relation with primary key and nonkey attributes.
Weak entity	Create a relation with a composite primary key (which includes the primary key of the entity on which this weak entity depends) and nonkey attributes.
Binary or unary 1:1 relationship	Place the primary key of either entity in the relation for the other entity or do this for both entities.
Binary 1:N relationship	Place the primary key of the entity on the one side of the relationship as a foreign key in the relation for the entity on the many side.
Binary or unary $M:N$ relationship or attributes associative entity	Create a relation with a composite primary key using the primary keys of the related entities, plus any nonkey of the relationship or associative entity.
Binary or unary $M:N$ relationship or associative entity with additional key(s)	Create a relation with a composite primary key using the primary keys of the related entities and additional primary key attributes associated with the relationship or associative entity, plus any nonkey attributes of the relationship or associative entity.
Binary or unary $M:N$ relationship or associative entity with its own key	Create a relation with the primary key associated with the relationship or associative entity, plus any nonkey attributes of the relationship or associative entity and the primary keys of the related entities (as foreign key attributes).



Transforming E-R Diagrams into Relations

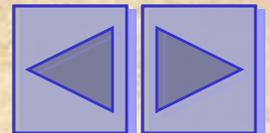
◆ Merging Relations

- Also called view integration
- The last step in the logical database design
- Some of the relations may be redundant; Purpose is to merge redundant relations
- View Integration Problems – when integrating relations, you must understand the meaning of the data and be prepared to resolve any problems that may arise in the process. The three problems are:
 - ◆ Synonyms
 - Two different names used for the same attribute
 - When merging, get agreement from users on a single, standard name



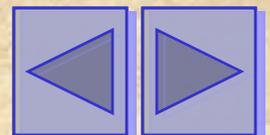
Transforming E-R Diagrams into Relations

- ◆ Homonyms
 - A single attribute name that is used for two or more different attributes
 - Resolved by creating a new name
- ◆ Dependencies between nonkeys
 - Dependencies may be created as a result of view integration
 - In order to resolve, the new relation must be normalized
- ◆ Class/subclass relationship may be hidden



Physical File and Database Design

- ◆ Designing physical files and databases requires following information:
 - 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



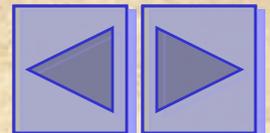
Designing Fields

◆ Field

- The smallest unit of named application data recognized by system software
- Each attribute from each relation will be represented as one or more fields

◆ Choosing data types

- Data Type
 - ◆ A coding scheme recognized by system software for representing organizational data
- 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
- Calculated fields
 - ◆ A field that can be derived from other database fields



Methods of Controlling Data Integrity

◆ Default Value

- A value a field will assume unless an explicit value is entered for that field

◆ Range Control

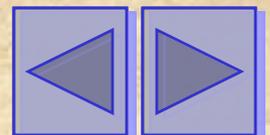
- Limits range of values which can be entered into field

◆ 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

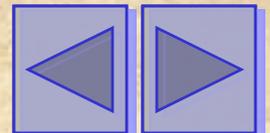
◆ Null Value

- A special field value, distinct from 0, blank, or any other value, that indicates that the value for the field is missing or otherwise unknown



Designing Physical Tables

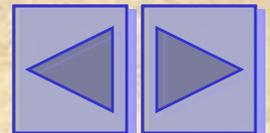
- ◆ Relational database is a set of related tables
- ◆ Physical Table
 - A named set of rows and columns that specifies the fields in each row of the table
- ◆ Design Goals
 - Efficient use of secondary storage (disk space)
 - ◆ Disks are divided into units that can be read in one machine operation
 - ◆ Space is used most efficiently when the physical length of a table row divides close to evenly with storage unit
 - Efficient data processing
 - ◆ Data are most efficiently processed when stored next to each other in secondary memory



Designing Physical Tables

◆ Denormalization

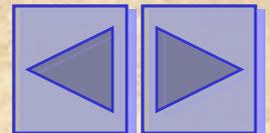
- The process of splitting or combining normalized relations into physical tables based on affinity of use of rows and fields
- Partitioning
 - ◆ Capability to split a table into separate sections
 - ◆ Oracle implements three types
 - Range
 - Hash
 - Composite
- Optimizes certain operations at the expense of others



Designing Physical Tables

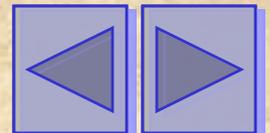
◆ Denormalization

- Three common situations where denormalization may be used
 1. Two entities with a one-to-one relationship
 2. A many-to-many relationship with nonkey attributes
 3. Reference data



Designing Physical Tables

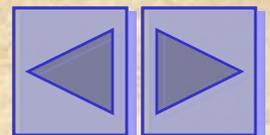
- ◆ Arranging Table Rows
 - Physical File
 - ◆ A named set of table rows stored in a contiguous section of secondary memory
 - Each table may be a physical file or whole database may be one file, depending on database management software utilized



Designing Physical Tables

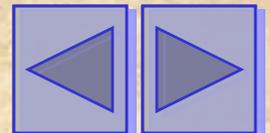
◆ File Organization

- A technique for physically arranging the set of table rows in secondary memory
- A computer operating system stores data in a physical file, which is a named set of table rows stored in a contiguous section of secondary memory.
- A file contains rows and columns from one or more tables, as produced from denormalization.



Designing Physical Tables

- Objectives for choosing file organization
 1. Fast data retrieval
 2. High throughput for processing transactions
 3. Efficient use of storage space
 4. Protection from failures or data loss
 5. Minimizing need for reorganization
 6. Accommodating growth
 7. Security from unauthorized use



Designing Physical Tables

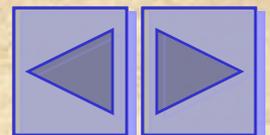
◆ Types of File Organization

■ Sequential

- ◆ The rows in the file are stored in sequence according to a primary key value
- ◆ Updating and adding records may require rewriting the file
- ◆ Deleting records results in wasted space

■ Indexed

- ◆ The rows are stored either sequentially or nonsequentially and an index is created that allows software to locate individual rows
- ◆ Index
 - A table used to determine the location of rows in a file that satisfy some condition
- ◆ Secondary Index
 - Index based upon a combination of fields for which more than one row may have same combination of values



Designing Physical Tables

- ◆ Guidelines for choosing indexes
 - Specify a unique index for the primary key of each table
 - Specify an index for foreign keys
 - Specify an index for nonkey fields that are referenced in qualification, sorting and grouping commands for the purpose of retrieving data
- Hashed File Organization
 - ◆ The address for each row is determined using an algorithm

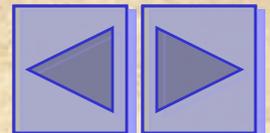
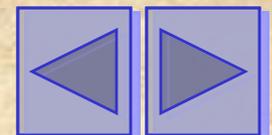


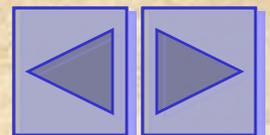
TABLE 12-3 Comparative Features of Sequential, Indexed, and Hashed File Organizations

<i>Factor</i>	<i>File Organization</i>		
	<i>Sequential</i>	<i>Indexed</i>	<i>Hashed</i>
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



Designing Controls for Files

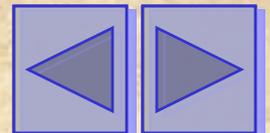
- ◆ Controls address protection from failures or data loss and security from unauthorized use
- ◆ File restoration techniques after damage or loss
 - periodically making a backup copy of a file,
 - storing a copy of each change to a file in a transaction log or audit trail, or
 - storing a copy of each row before or after it is changed.



Designing Controls for Files

◆ Data Security Techniques

- Coding or encrypting
- User account management
- Prohibiting users from working directly with the data. Users work with a copy which updates the files only after validation checks



Designing Forms and Reports

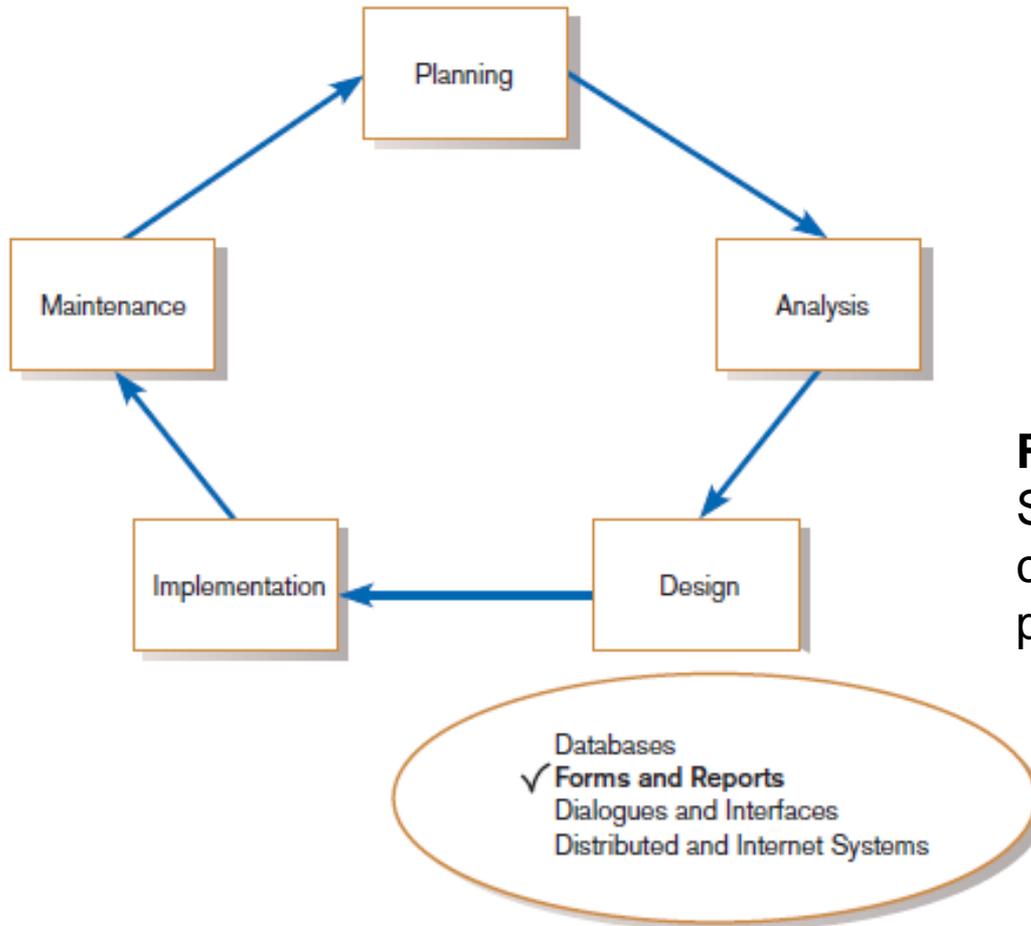


FIGURE 10-1
Systems development life cycle with logical design phase highlighted

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.

Designing Forms and Reports (Cont.)

- **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 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.



The Process of Designing Forms and Reports

- 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.



The Process of Designing Forms and Reports (Cont.)

- 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.)

The screenshot shows a window titled "Customer Information Entry" with a subtitle "Customer Information" and a date "Today: 2-Nov-10". The main form area is titled "CUSTOMER INFORMATION" and contains the following fields:

Customer Number:	1273
Name:	Contemporary Designs
Address:	123 Oak Street
City:	Austin
State:	TX
Zip:	28384

At the bottom of the form area are three buttons: "Save", "Help", and "Exit".

FIGURE 10-3

A data input screen designed in Microsoft's Visual Basic .NET



Deliverables and Outcomes

- Design specifications are the major deliverables and inputs to the system implementation phase.

Deliverables and Outcomes (Cont.)

- Design specifications have three sections:
 - *Narrative overview*: characterizes users, tasks, system, and environmental factors
 - *Sample design*: image of the form (from coding sheet or form building development tool)
 - *Testing and usability assessment*: measuring test/usability results (consistency, sufficiency, accuracy, etc.)



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



Formatting Forms and Reports (Cont.)

- *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)

(a) Poorly designed form

CUSTOMER INFORMATION		
CUSTOMER NO:	1273	
NAME:	CONTEMPORARY DESIGNS	
ADDRESS:	123 OAK ST.	
CITY-STATE-ZIP:	AUSTIN, TX 28384	
YTD-PURCHASE:	47,285.00	
CREDIT LIMIT:	10,000.00	
YTD-PAYMENTS:	42,656.65	
DISCOUNT %:	5.0	
PURCHASE:	21-JAN-10	22,000.00
PAYMENT:	21-JAN-10	13,000.00
PURCHASE:	02-MAR-10	16,000.00
PAYMENT:	02-MAR-10	15,500.00
PAYMENT:	23-MAY-10	5,000.00
PURCHASE:	12-JUL-10	9,285.00
PAYMENT:	12-JUL-10	3,785.00
PAYMENT:	1-SEP-10	5,371.65
STATUS:	ACTIVE	

Formatting Forms and Reports (Cont.)

Easy to read: clear, balanced layout

Clear title

Pine Valley Furniture

Pine Valley Furniture

Detail Customer Account Information

Page: 2 of 2

Today: 2-Nov-10

Customer Number: 1273

Name: Contemporary Designs

DATE	PURCHASE	PAYMENT	CURRENT BALANCE
01-Jan-08			0.00
21-Jan-08	(22,000.00)		(22,000.00)
21-Jan-08		13,000.00	(9,000.00)
02-Mar-08	(16,000.00)		(25,000.00)
02-Mar-08		15,500.00	(9,500.00)
23-May-08		5,000.00	(4,500.00)
12-Jul-08	(9,285.00)		(13,785.00)
12-Jul-08		3,785.00	(10,000.00)
21-Jul-08		5,371.65	(4,628.35)
YTD-SUMMARY	(47,285.00)	42,656.65	(4,628.35)

Help

Prior Screen

Exit

Summary of account information

Clear navigation information

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.

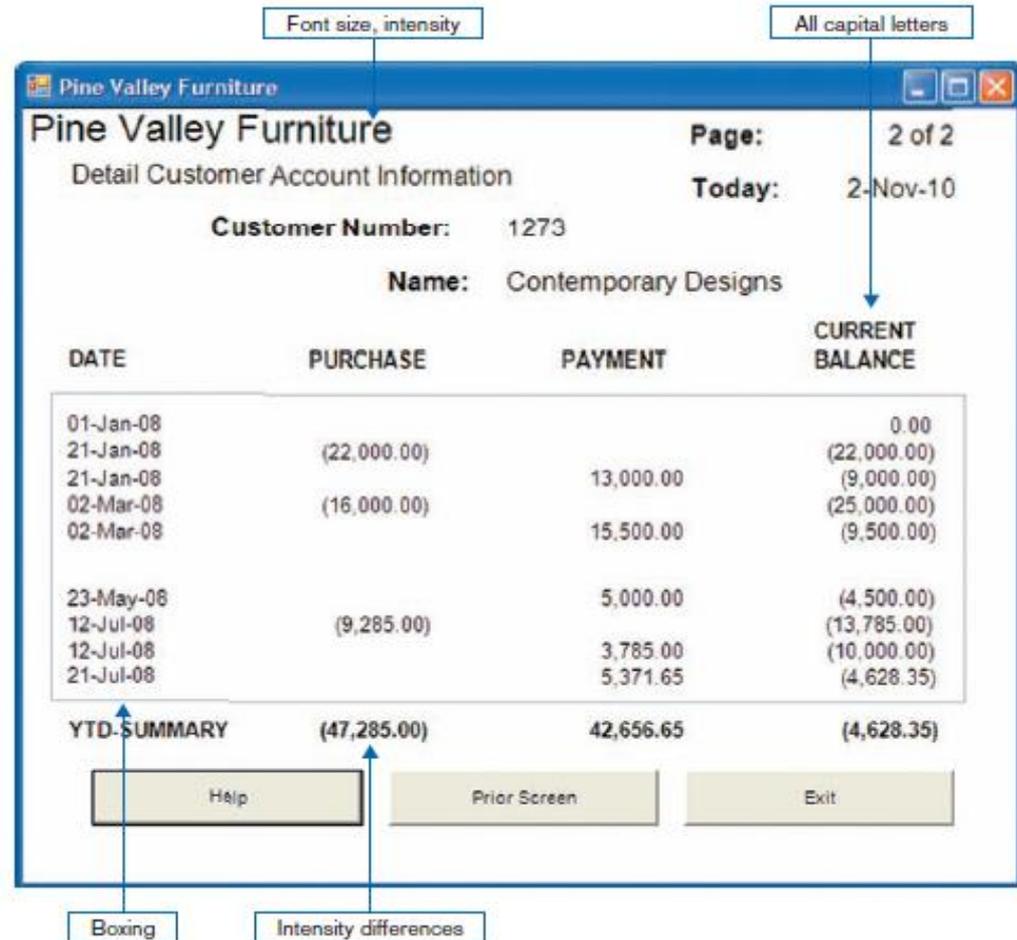
Highlighting Information (Cont.)

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)





Color vs. No Color

■ Benefits of Using 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.

Color vs. No Color (Cont.)

■ 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:** mixed upper and lower case, use conventional punctuation
- **Spacing:** double spacing if possible, otherwise blank lines between paragraphs
- **Justification:** left justify text, ragged right margins
- **Hyphenation:** no hyphenated words between lines
- **Abbreviations:** only when widely understood and significantly shorter than full text

Displaying Text (Cont.)

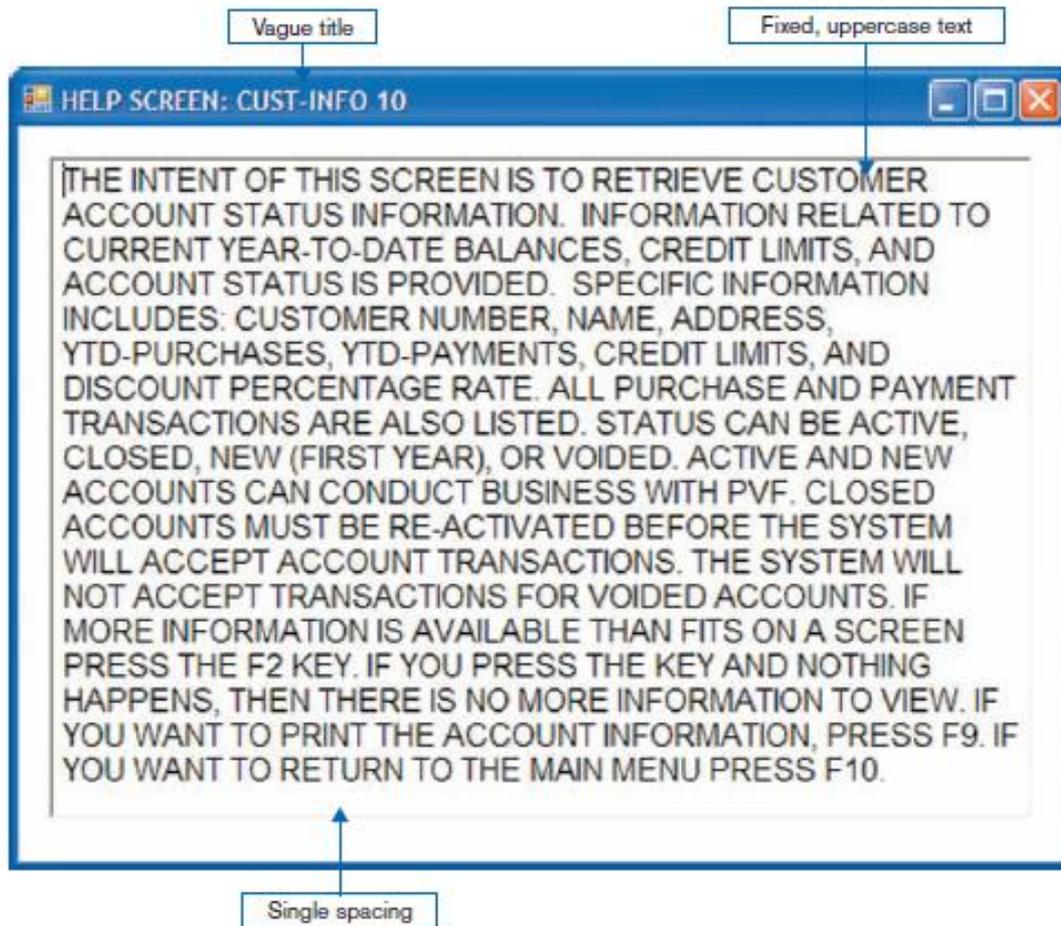


FIGURE 10-7

Contrasting the display of textual help information

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

Displaying Text (Cont.)

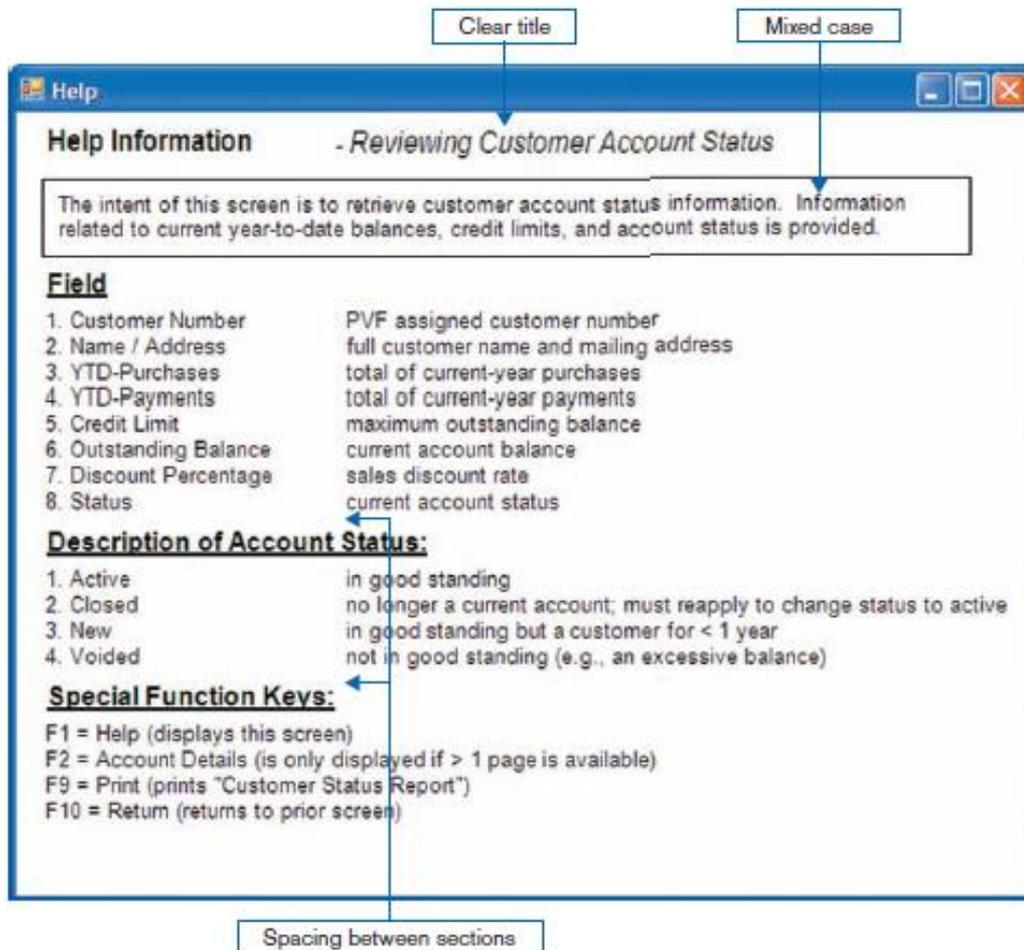


FIGURE 10-7 (continued)

(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.

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.



Designing Tables and Lists (Cont.)

- 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.)

- 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.)

Annotations:

- No column labels
- Single column for all types of data
- Numeric data are left justified

CUSTOMER INFORMATION		
CUSTOMER NO:	1273	
NAME:	CONTEMPORARY DESIGNS	
ADDRESS:	123 OAK ST.	
CITY-STATE-ZIP:	AUSTIN, TX 28384	
YTD-PURCHASE:	47,285.00	
CREDIT LIMIT:	10,000.00	
YTD-PAYMENTS:	42,656.65	
DISCOUNT %:	5.0	
PURCHASE:	21-JAN-10	22,000.00
PAYMENT:	21-JAN-10	13,000.00
PURCHASE:	02-MAR-10	16,000.00
PAYMENT:	02-MAR-10	15,500.00
PAYMENT:	23-MAY-10	5,000.00
PURCHASE:	12-JUL-10	9,285.00
PAYMENT:	12-JUL-10	3,785.00
PAYMENT:	1-SEP-10	5,371.65
STATUS:	ACTIVE	

FIGURE 10-8

Contrasting the display of tables and lists (Pine Valley Furniture)

(a) Poorly designed form

Designing Tables and Lists (Cont.)

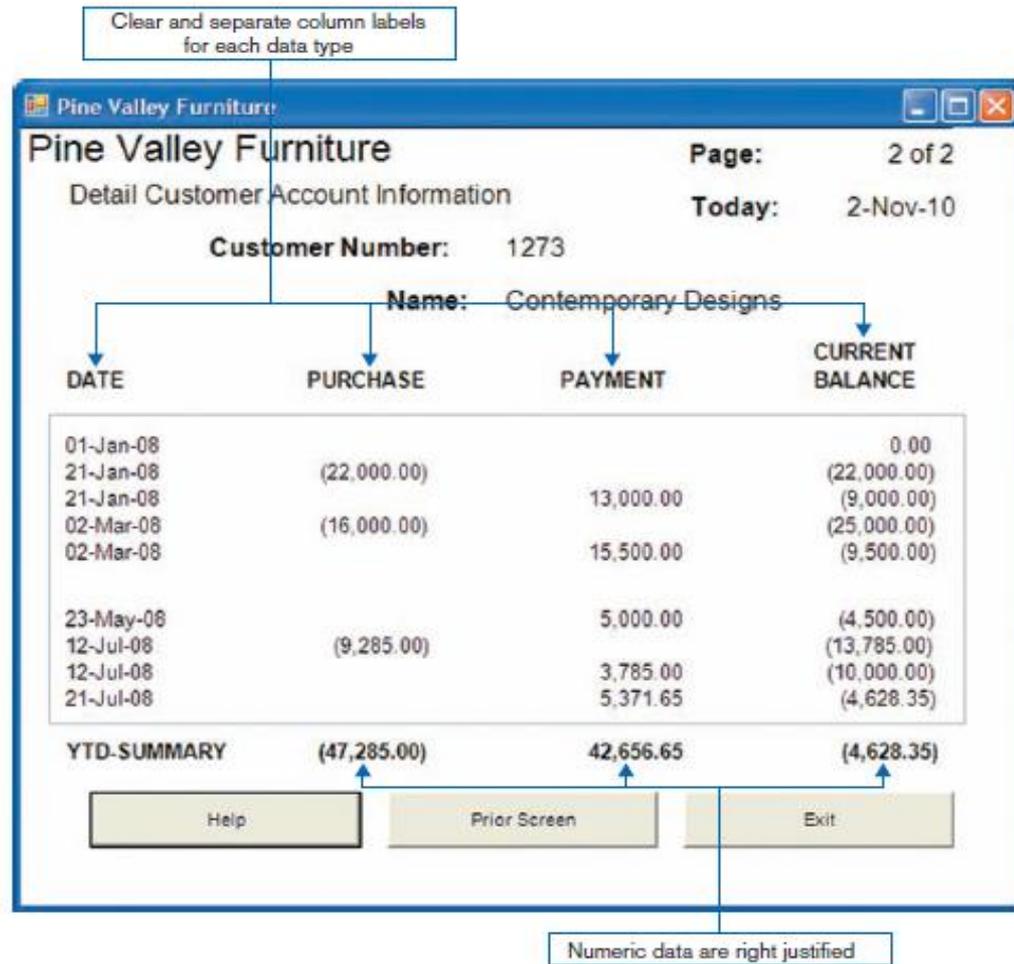


FIGURE 10-8 (continued)

(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.)

Place meaningful labels on all columns and rows

Alphabetic text is left justified

Use a meaningful title

Box the table data to improve the appearance of the table

Pine Valley Furniture
Salesperson Annual Summary Report, 2009

January 10, 2009 Page 1 of 2

Region	Salesperson	SSN	Quarterly Actual Sales			
			First	Second	Third	Fourth
Northwest & Mountain						
	Baker	999-99-9999	195,000	146,000	133,000	120,000
	Hawthorne	999-99-9999	220,000	175,000	213,000	198,000
	Hodges	999-99-9999	110,000	95,000	170,000	120,000
Midwest & Mid-Atlantic						
	Franklin	999-99-9999	110,000	120,000	170,000	90,000
	Stephenson ¹	999-99-9999	75,000	66,000	80,000	80,000
	Swenson	999-99-9999	110,000	98,000	100,000	90,000
New England						
	Brightman	999-99-9999	250,000	280,000	260,000	330,000
	Kennedy	999-99-9999	310,000	190,000	270,000	280,000

¹Sales reflect July 1, 2009 – December 31, 2009.

Superscript characters can be used to alert reader of more detailed information

Sort columns in some meaningful order (names are sorted alphabetically within region)

Long sequence of alphanumeric data is grouped into smaller segments

Right justify all numeric data

Try to fit table onto a single page to help in making comparisons

FIGURE 10-9
Tabular report illustrating numerous design guidelines

(Pine Valley Furniture)

Designing Tables and Lists (Cont.)

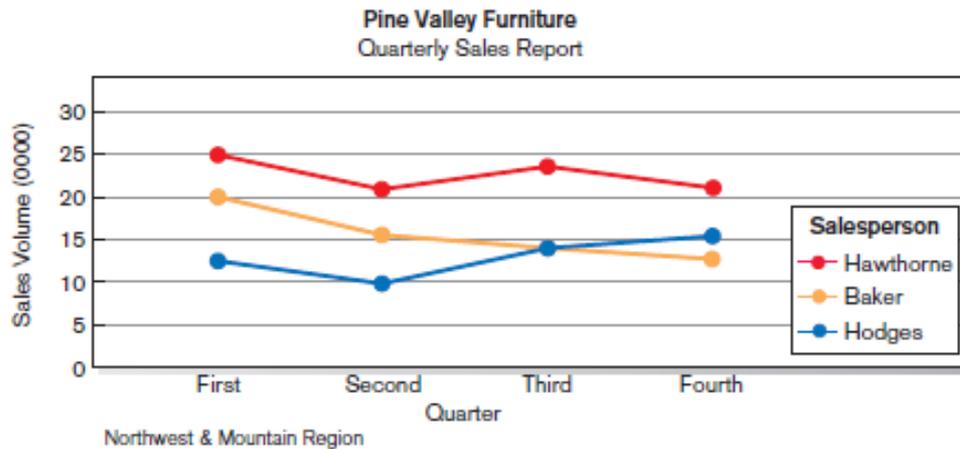
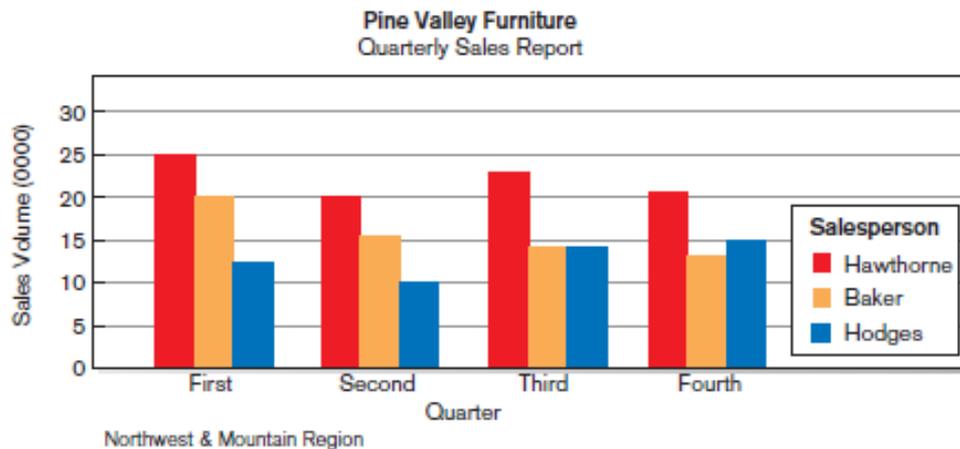


FIGURE 10-10
Graphs for comparison

(a) Line graph



(b) Bar graph

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?



Assessing Usability (Cont.)

- **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.

Usability Success Factors (Cont.)

- 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
- Layout of information should be consistent, both within and across applications



Measures of Usability (Cont.)

- Layout of information should be consistent both within and across applications, whether information is delivered on screen display or on a hard-copy report.



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.)

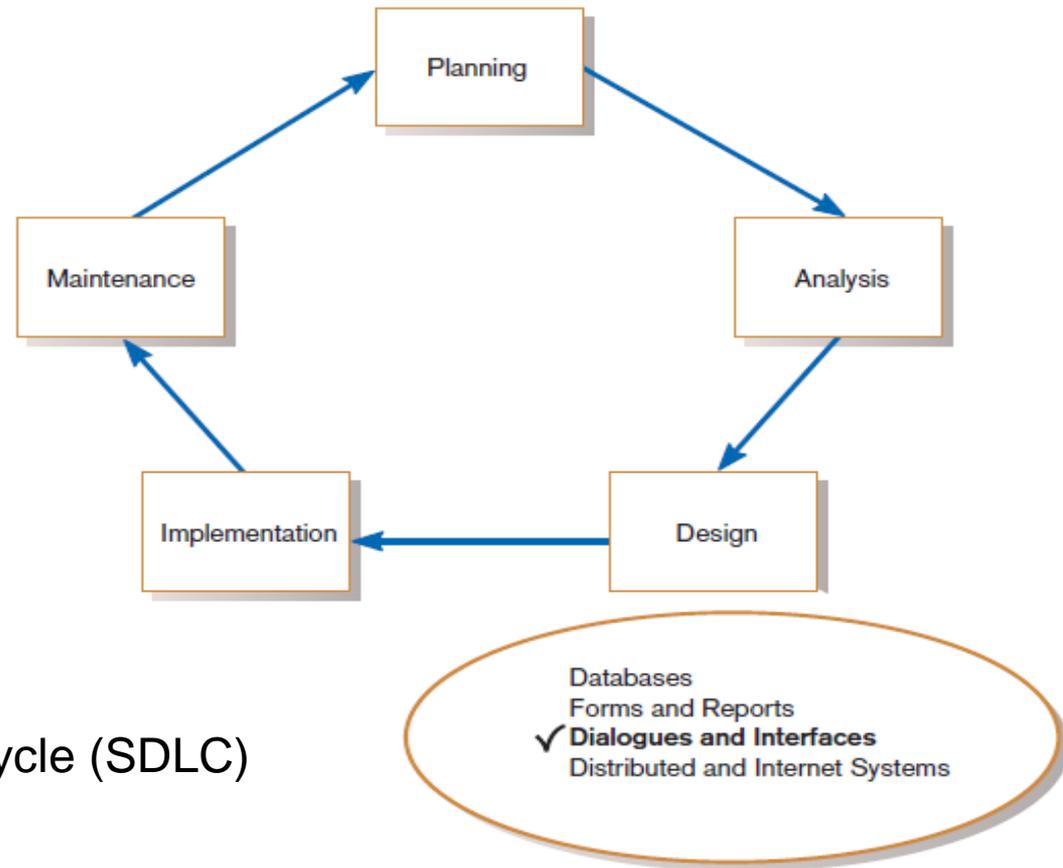


FIGURE 11-1
Systems development life cycle (SDLC)



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.

Deliverables and Outcomes (Cont.)

- The specification includes:
 - Narrative overview
 - Sample design
 - Testing and usability assessment
 - Dialogue sequence
- *Dialogue sequence* is the ways a user can move from one display to another.



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



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

Menu Interaction (Cont.)

- **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.

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.)

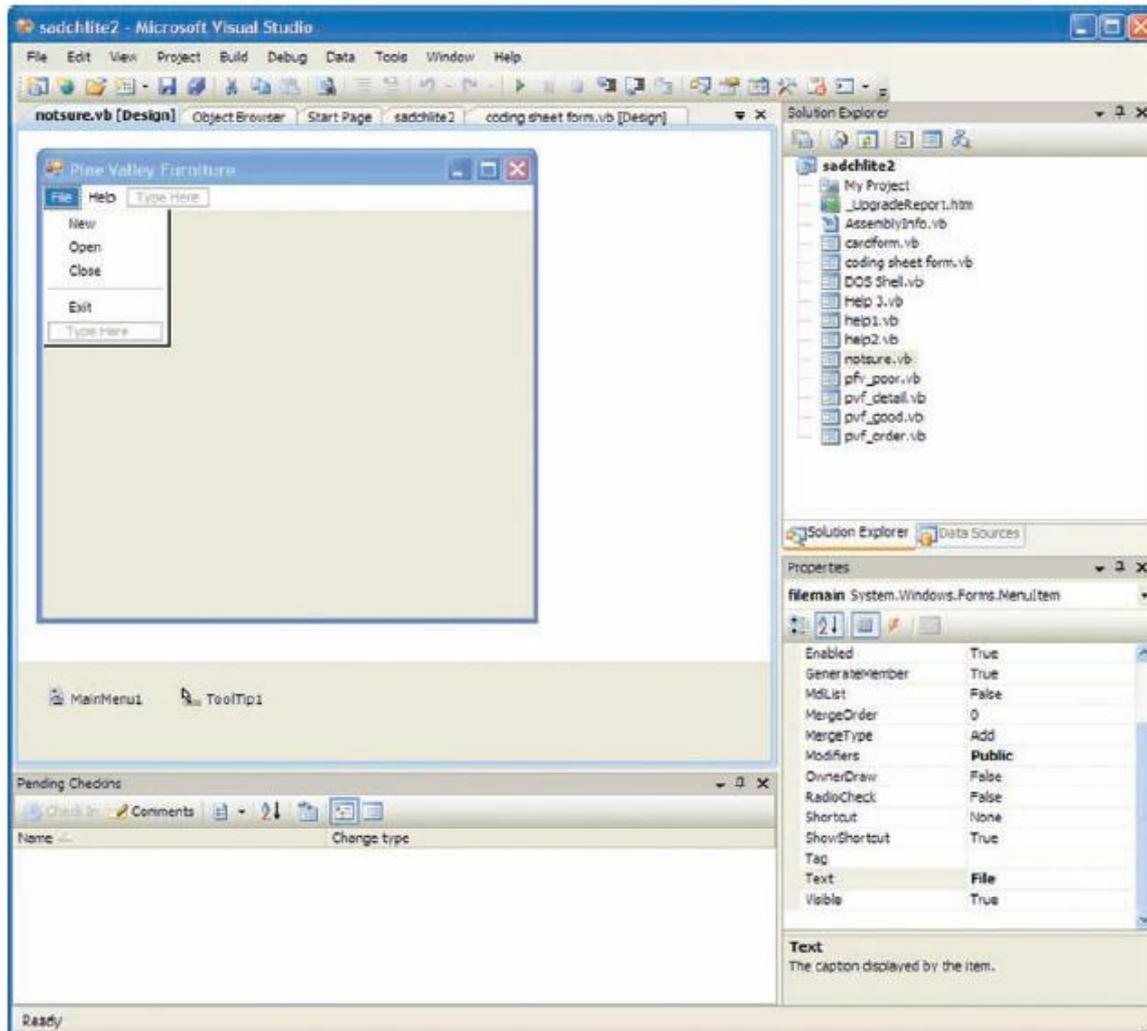


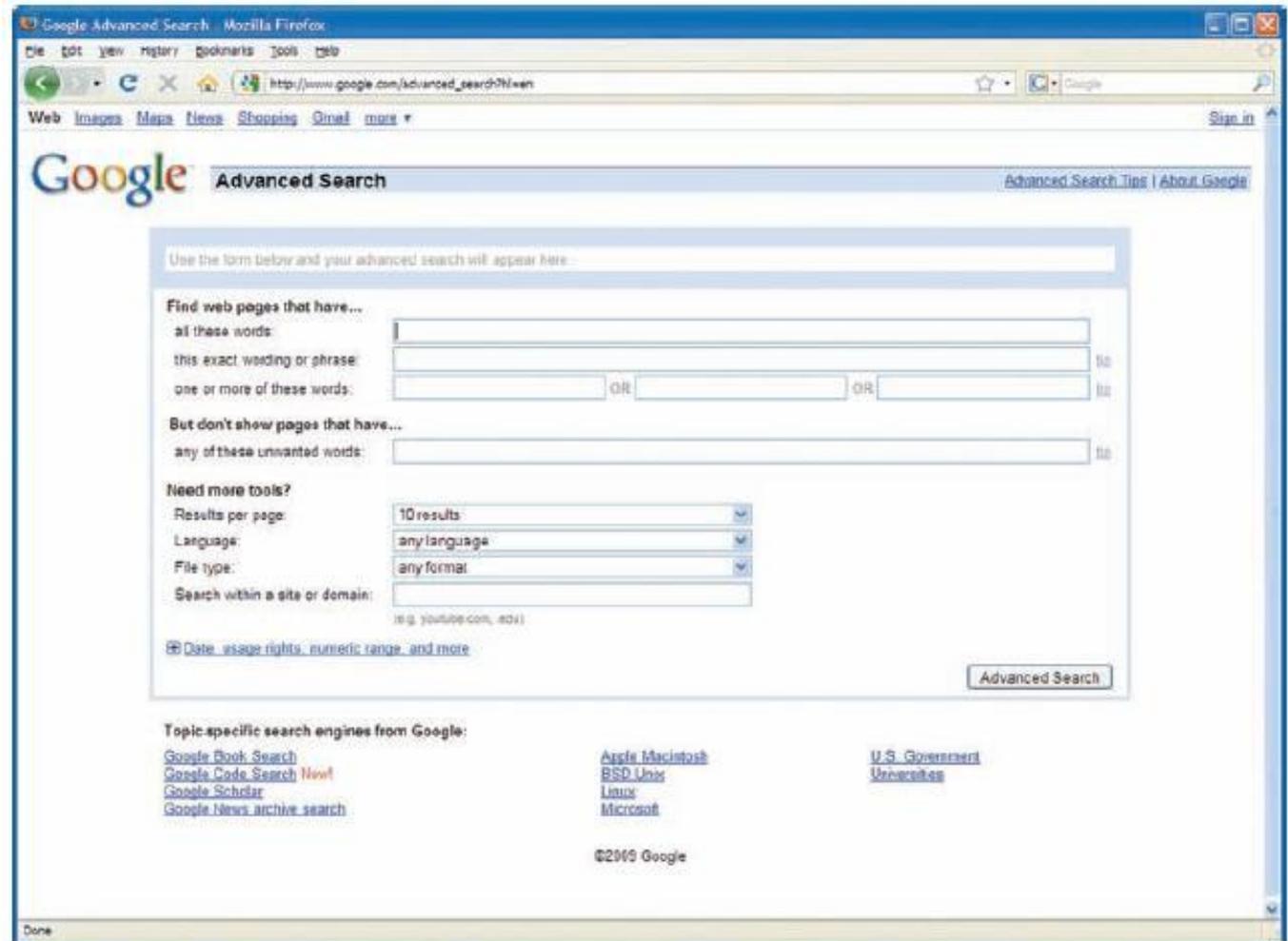
FIGURE 11-8
Menu building with
Microsoft Visual Basic
.NET

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.

Form Interaction (Cont.)

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



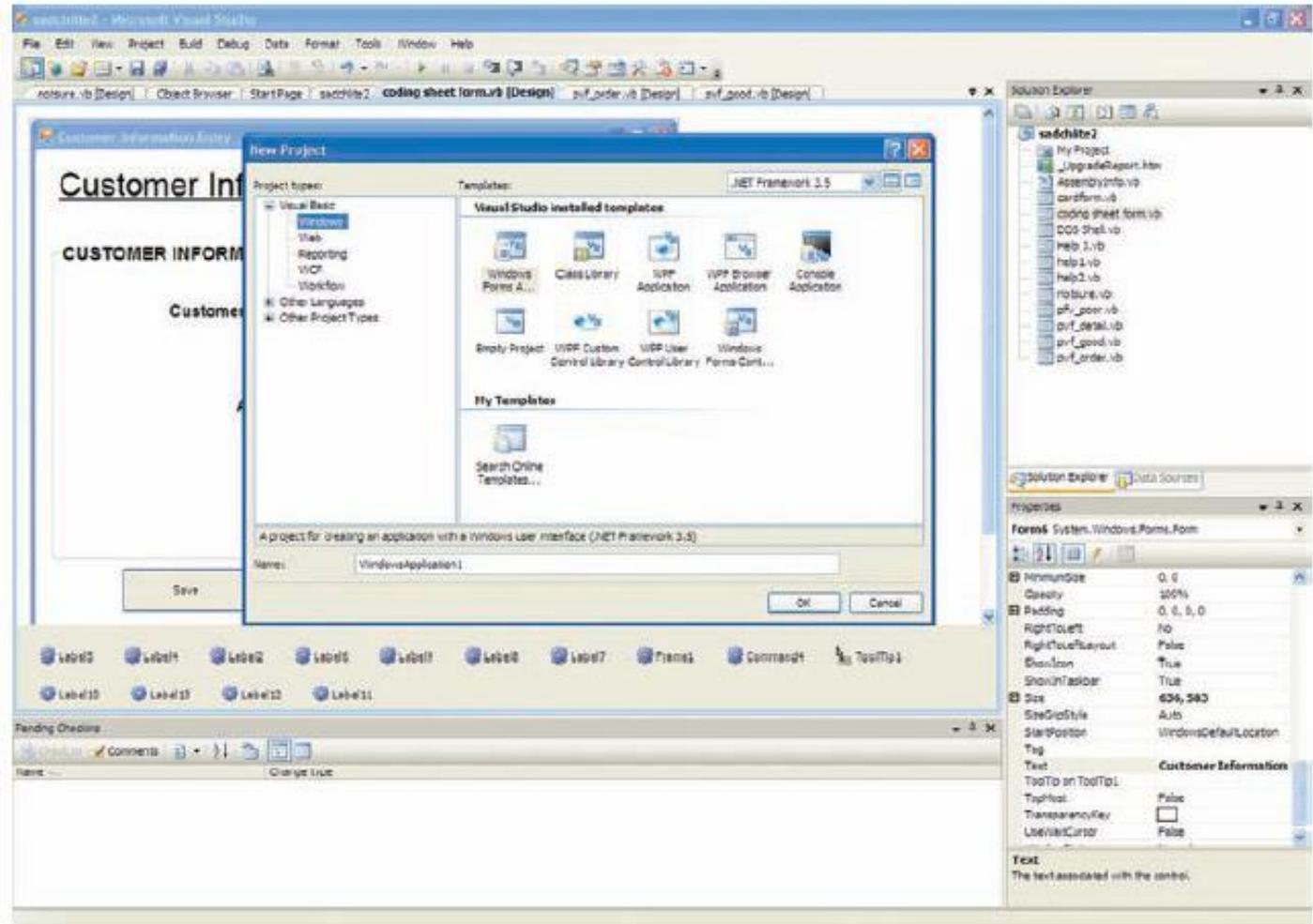


Object-Based Interaction

- **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

Object-Based Interaction (Cont.)

FIGURE 11-10
Object-based
(icon) interface
from Microsoft
Visual Basic
.NET





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.

Usability Problems with Hardware Devices

- Visual Blocking
 - touch screen, light pen
- User Fatigue
 - touch screen, light pen
- Movement Scaling
 - keyboard, mouse, joystick, trackball, graphics tablet, voice
- Durability
 - trackball, touch screen
- Adequate Feedback
 - keyboard, mouse, joystick, trackball, graphics tablet, voice
- Speed
 - keyboard
- Pointing Accuracy
 - joystick, touch screen, light pen, voice



Natural Language Interaction (Cont.)

- Usability problems with hardware devices:
 - Visual Blocking
 - touch screen, light pen
 - User Fatigue
 - touch screen, light pen



Natural Language Interaction (Cont.)

■ Movement Scaling

- keyboard, mouse, joystick, trackball, graphics tablet, voice

■ Durability

- trackball, touch screen

Natural Language Interaction (Cont.)

■ Adequate Feedback

- keyboard, mouse, joystick, trackball, graphics tablet, voice

■ Speed

- keyboard

■ Pointing Accuracy

- joystick, touch screen, light pen, voice

Natural Language Interaction (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

(Source: Based on Blattner & Schultz, 1988.)

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



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



Designing Interfaces (Cont.)

- Use standard formats similar to paper-based forms and reports.
- Use left-to-right, top-to-bottom navigation.



Designing Interfaces (Cont.)

- 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 data entry errors in a field:
 - Appending: adding additional characters
 - Truncating: losing characters
 - Transcribing: entering invalid data
 - Transposing: reversing sequence of characters

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.



Designing Dialogues

- **Dialogues:** 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.

Designing the Dialogue Sequence (Cont.)

- **Dialogue diagramming:** a formal method for designing and representing human-computer dialogues using box and line diagrams

Designing the Dialogue Sequence (Cont.)

- Three sections of the box are used as:
 - *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.

Designing the Dialogue Sequence (Cont.)

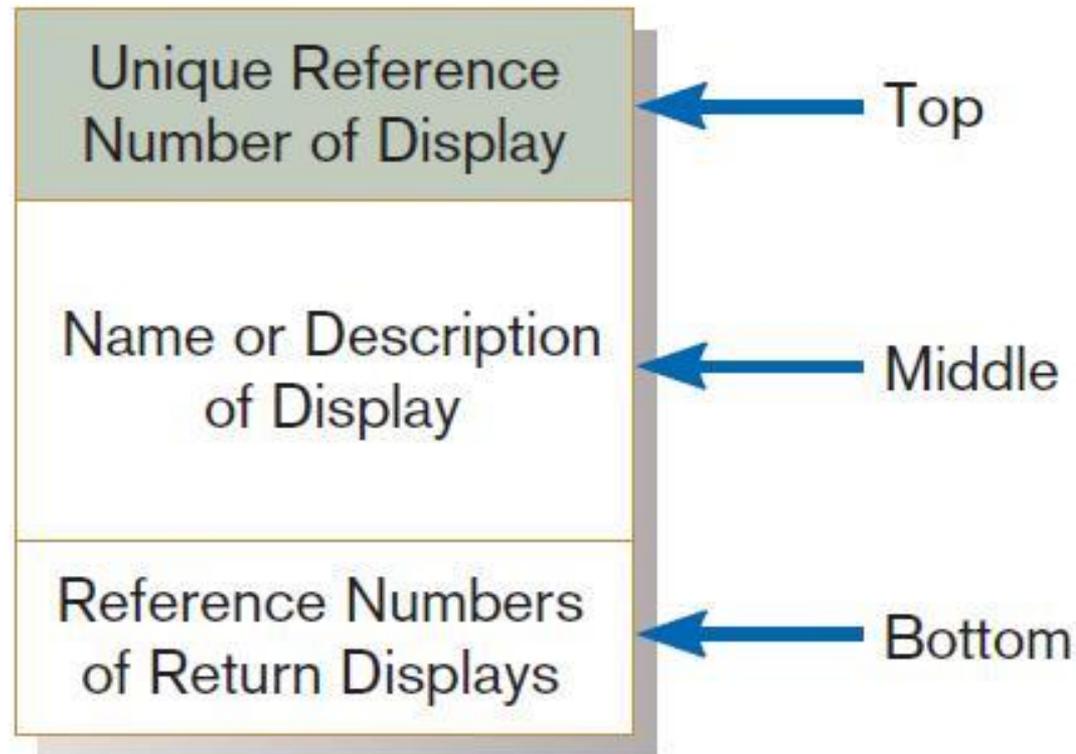


FIGURE 11-17
Sections of a dialogue diagramming box

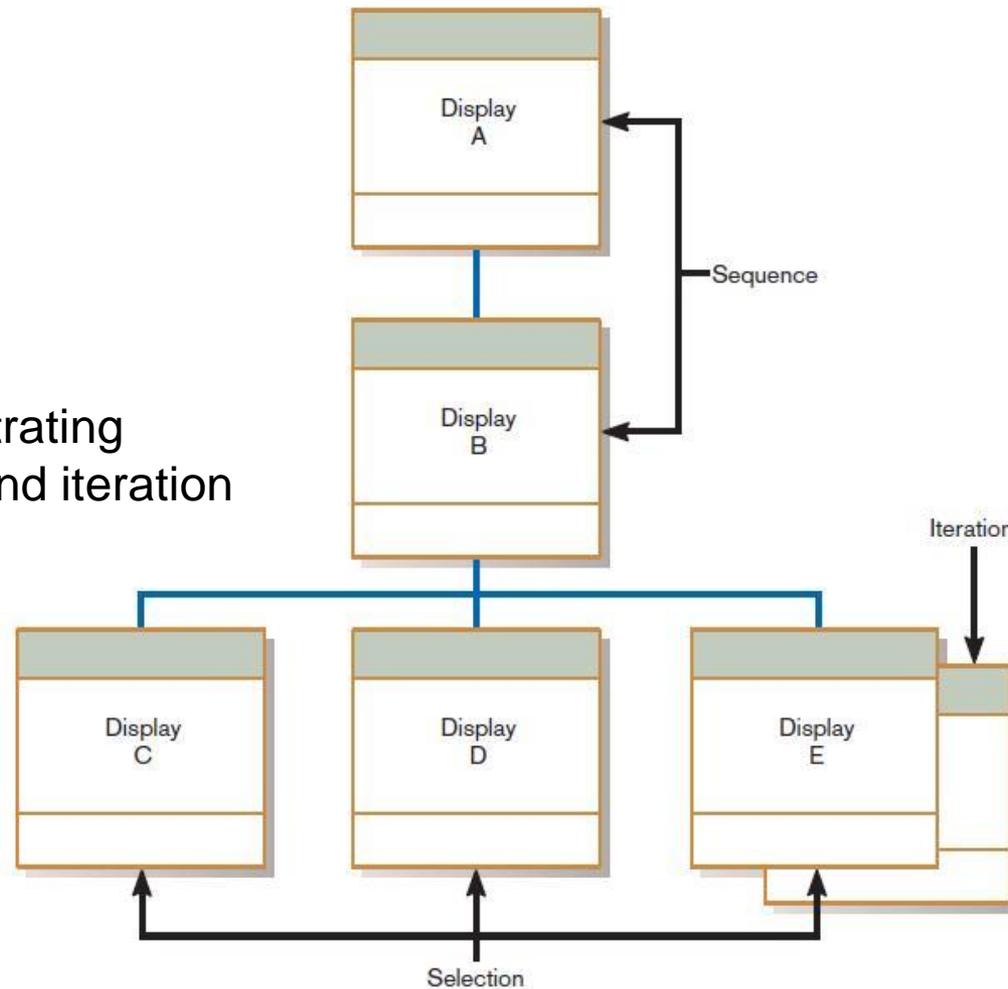


Designing the Dialogue Sequence (Cont.)

- 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



Building Prototypes and Assessing Usability

- Optional activities
- Building prototypes displays using a graphical development environment
 - Microsoft's Visual Studio .NET
 - Borland's Enterprise Studio
 - Easy-to-use input and output (form, report, or window) design utilities



Graphical Interface Design Issues

- Become an expert user of the GUI environment.
 - Understand how other applications have been designed.
 - Understand standards.
- Gain an understanding of 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 graphical
user interface design
standards

