USN | | | | | | | | | | |

Internal Assessment Test II – Dec 2022

| Sub: | USER INTERFACE DESIGN | | | | | Sub Code: | 18CS734 | Branch: | CSE |
|---|---|---|---|---|---|---|---|---|---|
| Date: | 2/12/22 | Duration: | 90 mins | Max Marks: | 50 | Sem/Sec: | VII A,B,C | | OBE |

Answer any FIVE FULL Questions

| | | MARKS | CO | RBT |
|---|---|---|---|---|
| 1. | **Explain the different types of graphical menus.** | [6+4] | CO3 | L3 |

**1. Menu Bar**
**2. Pull-Down Menu**
**3. Cascading Menus**
**4. Pop-up Menus**
**5. Tear-off Menus**
**6. Iconic Menus**
**7. Pie Menus**
**8. Default Menu Items**
**9. Functions Not Represented by Default Items**

**Menu Bar** — To identify and provide access to common and frequently used application actions that take place in a wide variety of different windows.
— A menu bar choice by itself should not initiate an action.


**Figure 4.19** Menu bar composed of text.


**Figure 4.20** Menu bar composed of buttons.

The *advantages* of menu bars are that they:
- Are always visible, reminding the user of their existence.
- Are easy to browse through.
- Are easy to locate consistently on the screen.

The *disadvantages* of menu bars are that:
- They consume a full row of screen space.
- They require looking away from the main working area to find.
- They require moving pointer from the main working area to select.

## Pull-Down Menu

To initiate frequently used application actions that take place on a wide variety of different windows.
— A small number of items.
— Items best represented textually.
— Items whose content rarely changes.

The ***advantages*** of pull-down menus are:
- The menu bar cues a reminder of their existence.
- They may be located relatively consistently on the screen.

 No window space is consumed when they are not used.

The ***disadvantages*** of pull-down menus are:

 They require searching and selecting from another menu before seeing options.

 They require looking away from main working area to read.

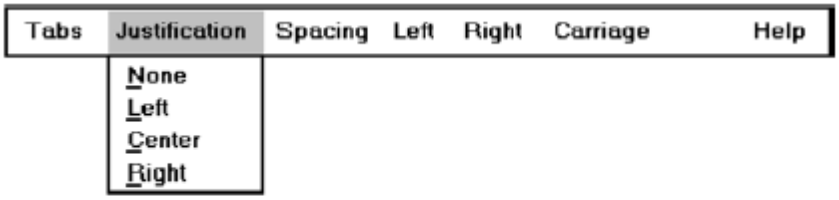



**Figure 4.24** Menu bar pull-down.

## Cascading Menus

To reduce the number of choices presented together for selection (reduce menu breadth).

— When a menu specifies many alternatives and the alternatives can be grouped in meaningful related sets on a lower-level menu.

— When a choice leads to a short, fixed list of single-choice properties.

— When there are several fixed sets of related options.

— To simplify a menu.

— Avoid using for frequent, repetitive commands.

The ***advantages*** of cascading menus are that:

The top-level menus are simplified because some choices are hidden.

 More first-letter mnemonics are available because menus possess fewer alternatives.

 High-level command browsing is easier because subtopics are hidden.

The ***disadvantages*** of cascading menus are:

 Access to submenu items requires more steps.

 Access to submenu items requires a change in pointer movement direction.

 Exhaustive browsing is more difficult; some alternatives remain hidden as pulldowns become visible

## Pop-up Menus

Use to present alternatives or choices within the context of the task.

The ***advantages*** of pop-up menus are:

 They appear in the working area.

 They do not use window space when not displayed.

 No pointer movement is needed if selected by button.

The ***disadvantages*** of pop-up menus are:

 Their existence must be learned and remembered.

 Means for selecting them must be learned and remembered.

 They require a special action to see the menu (mouse click).

## Tear-off Menus

**Advantages/disadvantages.** No space is consumed on the screen when the menu is not needed. When needed, it can remain continuously displayed. It does require extra steps to retrieve, and it may obscure the screen working area.

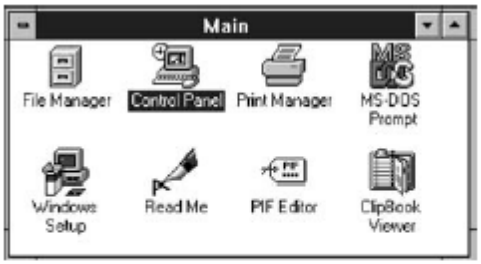


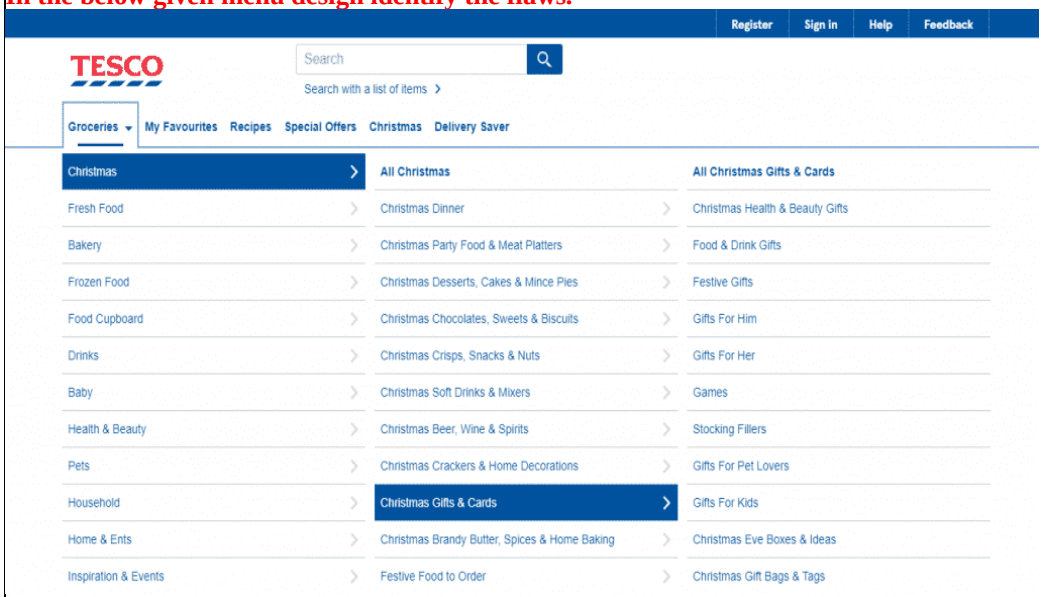

**Figure 4.36** Iconic menu (from Microsoft Windows).

**Pie Menus**

Mouse-driven selections, with one- or two-level hierarchies, short lists, and choices conducive to the format.

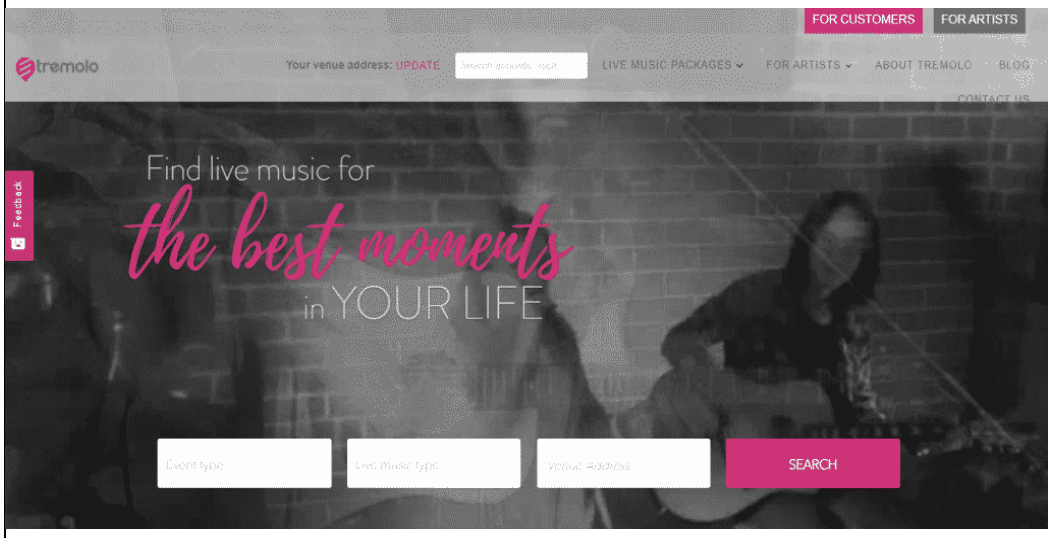
A pie menu is a circular representation of menu items, as illustrated in Figure 4.37, that can be used as an alternative to a pull-down or pop-up menu. Research has found that this style of menu yields higher performance than the typical vertical array, especially when the menu tasks are unrelated. Their basic advantage is that, when presented with the mouse pointer positioned in the pie's center, average movement to any pie wedge is shorter. Mayhew (1992) concludes that pie menus might work well for mouse-driven selections with one- or two-level hierarchies, short choice listings, and data conducive to the format. Performance advantages for keyboard selection are doubtful, however.

**In the below given menu design identify the flaws.**




- More than 8 choices in menu they are large
- Has almost three levels and continues
- No proper ordering of menus(nor as per sequence)
- No contrast to pull down or sub menu

| | | | | |
|---|---|---|---|---|
| 2(A) | **What care should be taken while formatting and phrasing menus.**<br>The human-computer interface has a rich history of experimental studies with menus, the results of which can and have been applied to graphical screen and Web page menu design and presentation. The guidelines for formatting menus are:<br>**1.Consistency** **[ Note- Brief explanation of each point is required]**<br>**2. Display** | [6] | CO3 | L2 |

| | | | | |
|---|---|---|---|---|
| | **3. Presentation**<br>**4. Organization**<br>**5. Complexity**<br>**6. Item Arrangement**<br>**7. Ordering**<br>**8. Grouping**<br>**9. List separators**<br><br>**Phrasing the Menu**<br>A menu must communicate to the user information about: The nature and purpose of the menu itself. The nature and purpose of each presented choice. How the proper choice or choices may be selected.<br>**1. Menu Titles**<br>**2. Menu Choice Descriptions**<br>**3. Menu Instructions**<br>**4. Intent Indicators**<br>**5. Keyboard Equivalents**<br>**6. Keyboard Accelerators** | | | |
| 2(B) | In the below design identify the basic mistakes in formatting and phrasing<br>  <br>• No Consistency in font size<br>• Color effects are very bad<br>• Fonts are not distinctive on button<br>• The Web site has no name | [4] | CO3 | L3 |
| 3 | **Explain the different structures for menu design**<br><br>**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.<br>1. Single Menus<br>2. Sequential Linear Menus | [10] | CO3 | L2 |

3. Simultaneous Menus
4. Hierarchical Menus
5. Connected Menus
6. Event-Trapping Menus

## 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 4.1.
In using the Internet, for example, at a point in the dialog people may be asked if they wish **to "Stay Connected" or "Disconnect."** In playing a game, choices presented may be ―novice,‖ ―intermediate,‖ or ―expert.‖ Single menus conceptually require choices from this single menu only, and no other menus will follow necessitating additional user choices.
**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.

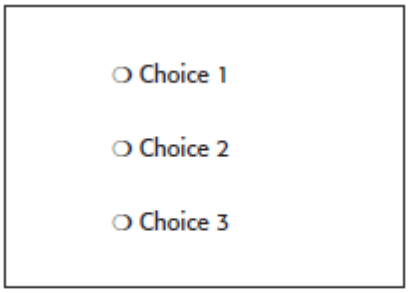



**Figure 4.1** Single menu.

## 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. A sequential linear menu is illustrated in Figure 4.2.

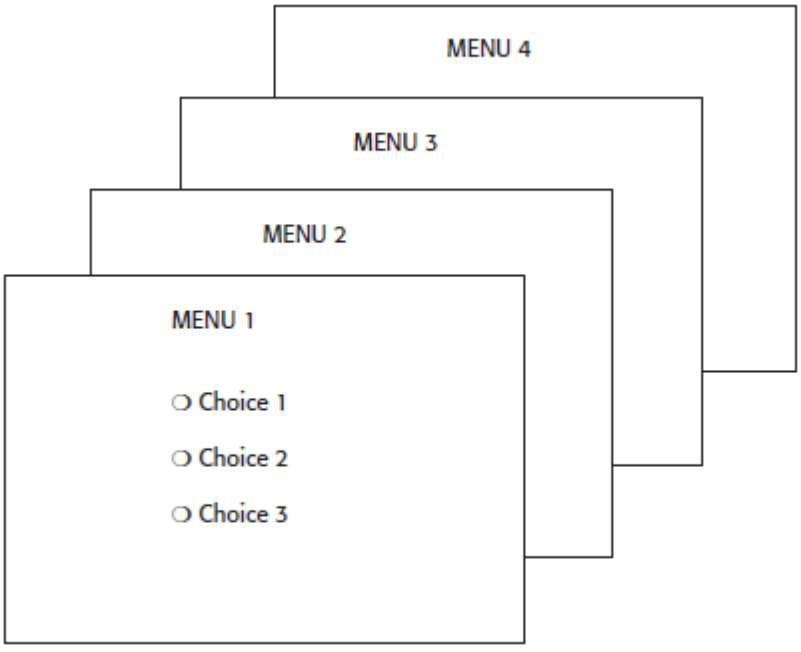



**Figure 4.2** Sequential linear menus.

### Disadvantages:

**A long sequence may become tedious as menu after menu is presented. The user may not remember an answer to a previous question, a question important to the currently presented choices.** The user may also want to return to a previous menu to change an answer or

look at an answer, an awkward process that must be allowed.

## 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. The tedium associated with a long series of sequential menus is greatly reduced.

**Disadvantages**:

Problems with simultaneous menus are that for large collections of menu alternatives **screen clutter can easily occur,** and screen paging or scrolling may still be necessary to view all the choices.

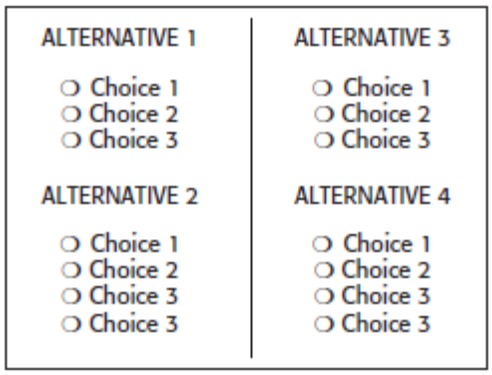



**Figure 4.3** Simultaneous menus.

## Hierarchical Menus

When many relationships exist between menu alternatives, and some menu options are only appropriate depending upon a previous menu selection, a hierarchical structure is the best solution. 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.

**Disadvantages:**

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

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.

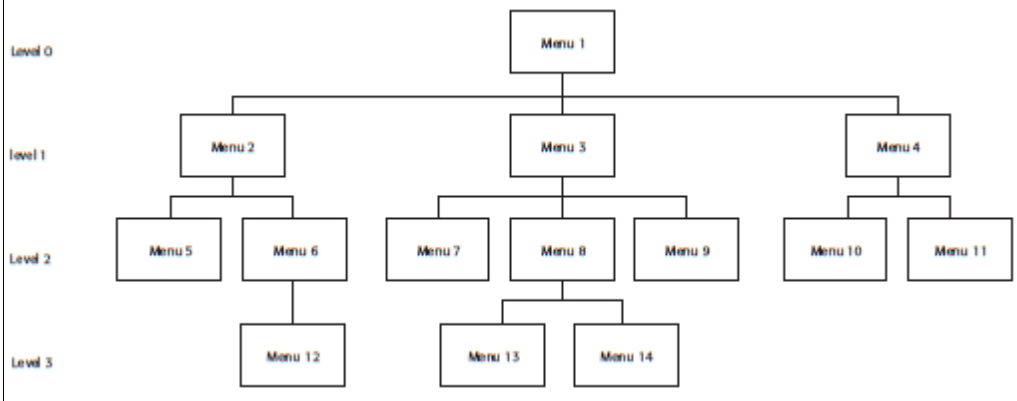



**Figure 4.4** Hierarchical menus.

## 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 acyclical, with movement permitted in only one direction.
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.

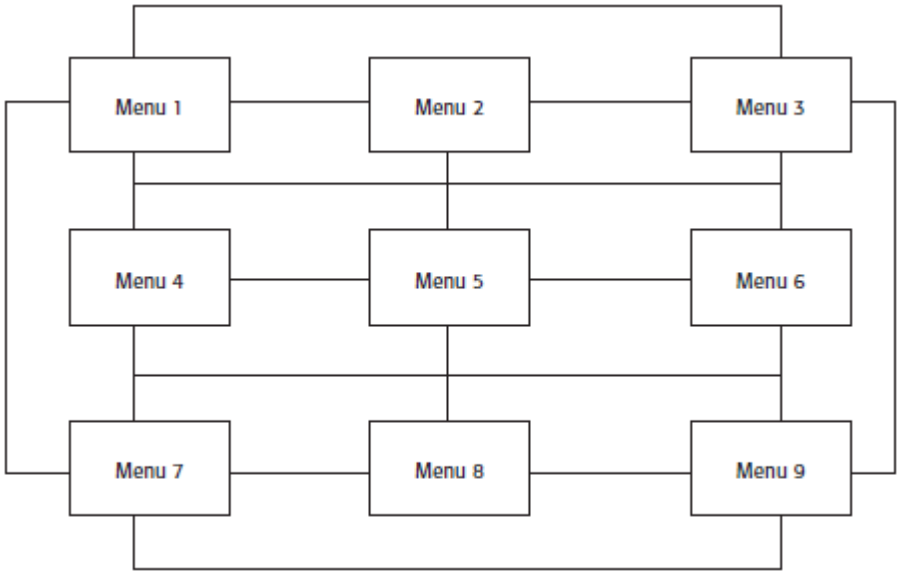


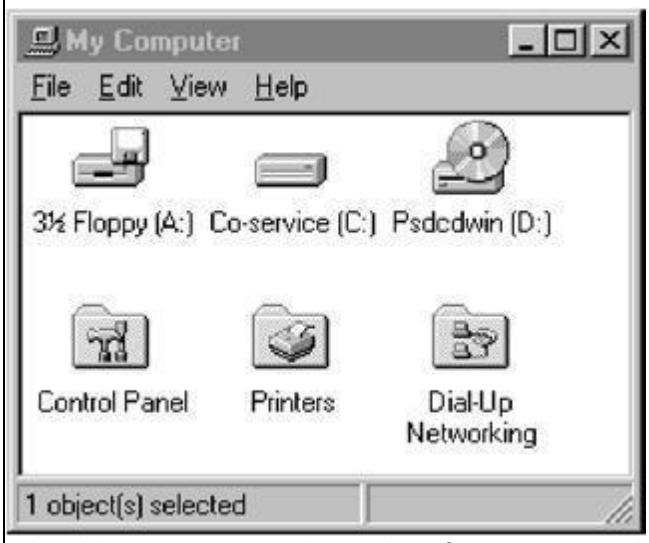

**Figure 4.5** Connected menus.

## 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).**

| | | | | |
|---|---|---|---|---|
| 4. | **With neat diagrams explain the different types of window designs.**<br>Types of windows are<br>1.Primarywindows<br>2.Secondarywindows<br>• Dependent and independent window<br>• Model and modeless<br>• Cascading and unfolding<br>• Dialog box and property inspectors<br>• Message box<br>• Palette Windows<br>• vii. Popup window<br><br>Primary windows | [10] | CO2 | L3 |




Should represent an independent function or application.
Use to present constantly used window components and controls.
• Menu bar items that are:
Used frequently.
Used by most, or all, primary or secondary windows.
• Controls used by dependent windows.
Use for presenting information that is continually updated.
• For example, date and time.
Use for providing context for dependent windows to be created.
Do not:
• Divide an independent function into two or more primary windows.
• Present unrelated functions in one primary window.

Secondary window


For performing subordinate, supplemental, or ancillary actions that are:
• Extended or more complex in nature.
• Related to objects in the primary window.
For presenting frequently or occasionally used window components.
■ Important guidelines:
Should typically not appear as an entry on the taskbar.
A secondary window should not be larger than 263 dialog units x 263 dialog units.
Secondary windows are supplemental windows. Secondary windows may be

dependent upon a primary window or displayed independently of the primary window. They structurally resemble a primary window, possessing some of the same action controls (Close button) and possibly a What's This? button.

A *dependent* secondary window is one common type. It can only be displayed from a command on the interface of its primary window. It is typically associated with a single data object, and appears on top of the active window when requested. It is movable, and scrollable. If necessary, it uses the primary window's menu bar. Most systems permit the use of multiple secondary windows to complete a task. In general, dependent secondary windows are closed when the primary window closes, and hidden when their primary window is hidden or minimized.

An *independent* secondary window can be opened independently of a primary window—for example, a property sheet displayed when the user clicks the Properties command on the menu of a desktop icon. An independent secondary window can typically be closed without regard to the state of any primary window unless there is an obvious relationship to the primary window.

Model-prompt and modeless-word search

■ Modal:

Use when interaction with any other window must not be permitted.

Use for:

Presenting information.

For example, messages (sometimes called a message box).

Receiving user input.

For example, data or information (sometimes called a prompt box).

Asking questions.

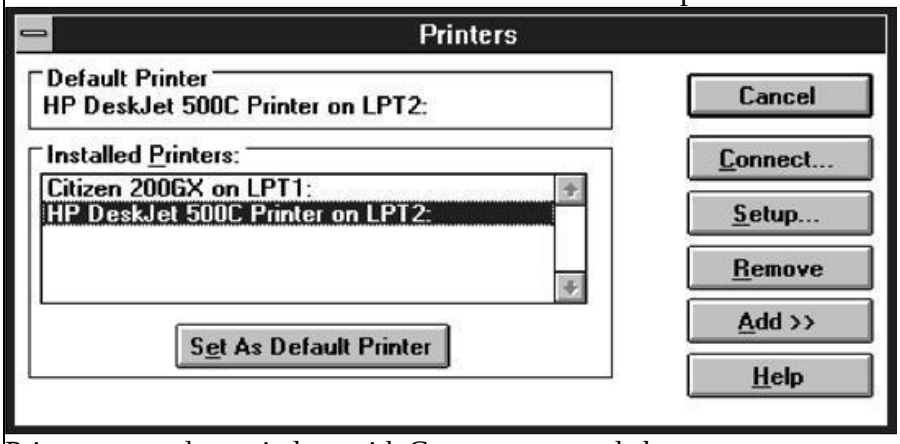
For example, data, information, or directions (sometimes called a question box).

Use carefully because it constrains what the user can do.

■ Modeless:

Use when interaction with other windows must be permitted.

Use when interaction with other windows must be repeated.




Printers secondary window with Connect... cascade button

Cascading:

Purpose:

To provide advanced options at a lower level in a complex dialog.

Guidelines:

Provide a command button leading to the next dialog box with a "To a Window" indicator, an ellipsis (. . . ).

- Present the additional dialog box in cascaded form.
- Provide no more than two cascades in a given path.
- Do not cover previous critical information.

Title Bar.
Relevant displayed information.
• If independent, close the secondary window from which it was opened.

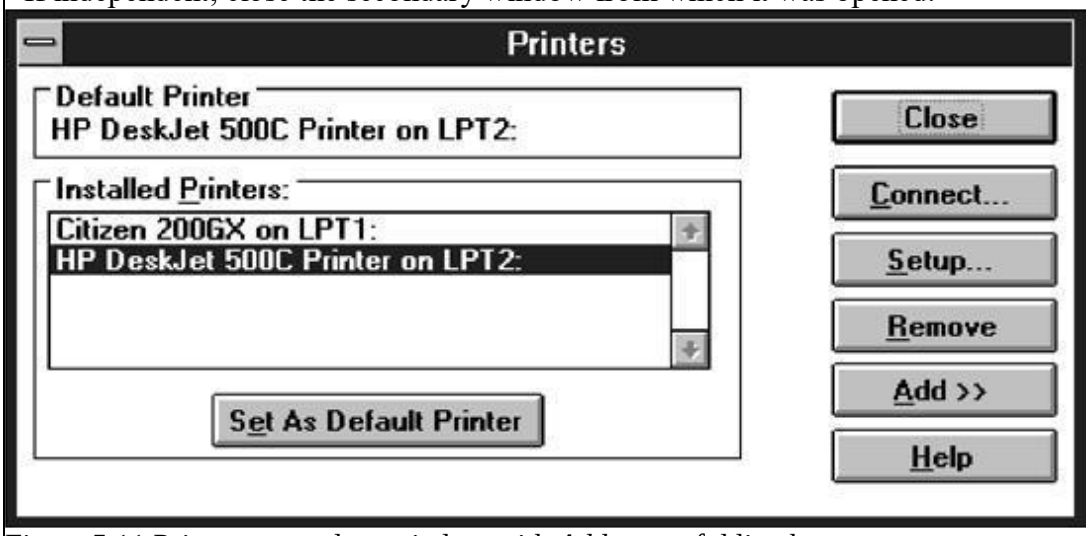



Figure 5.11 Printers secondary window with *Add >>* unfolding button.

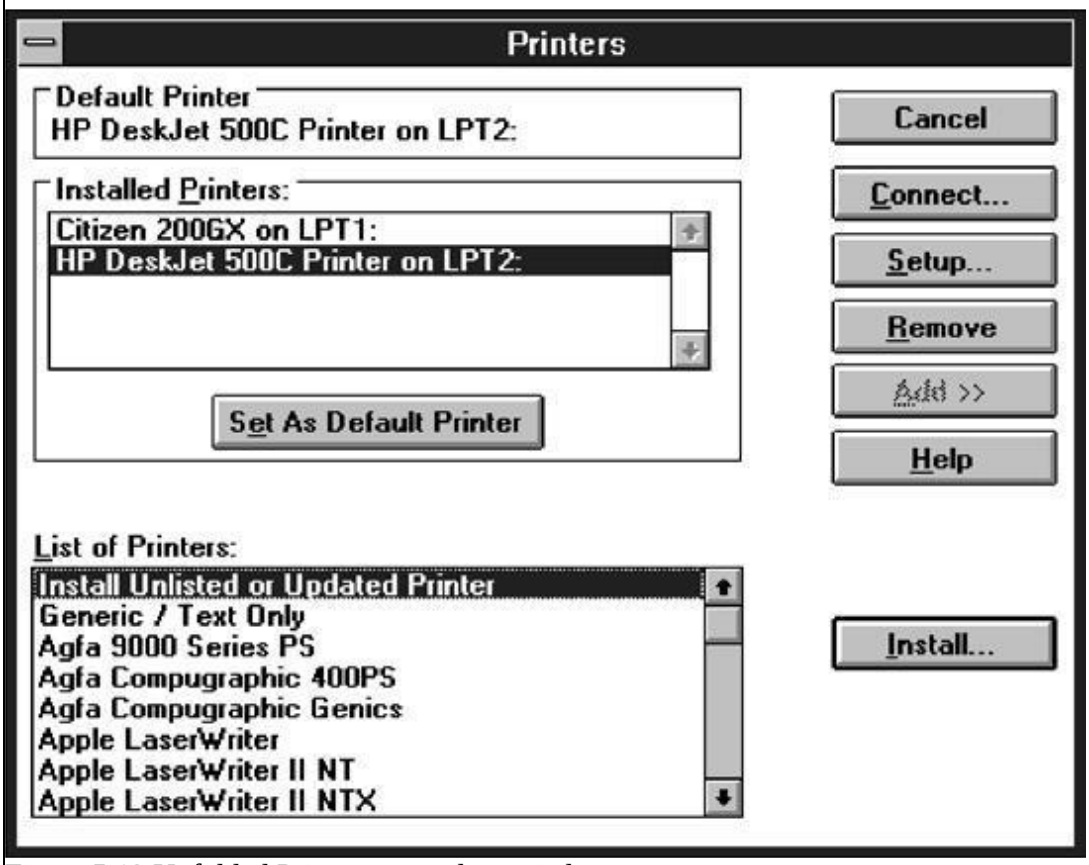
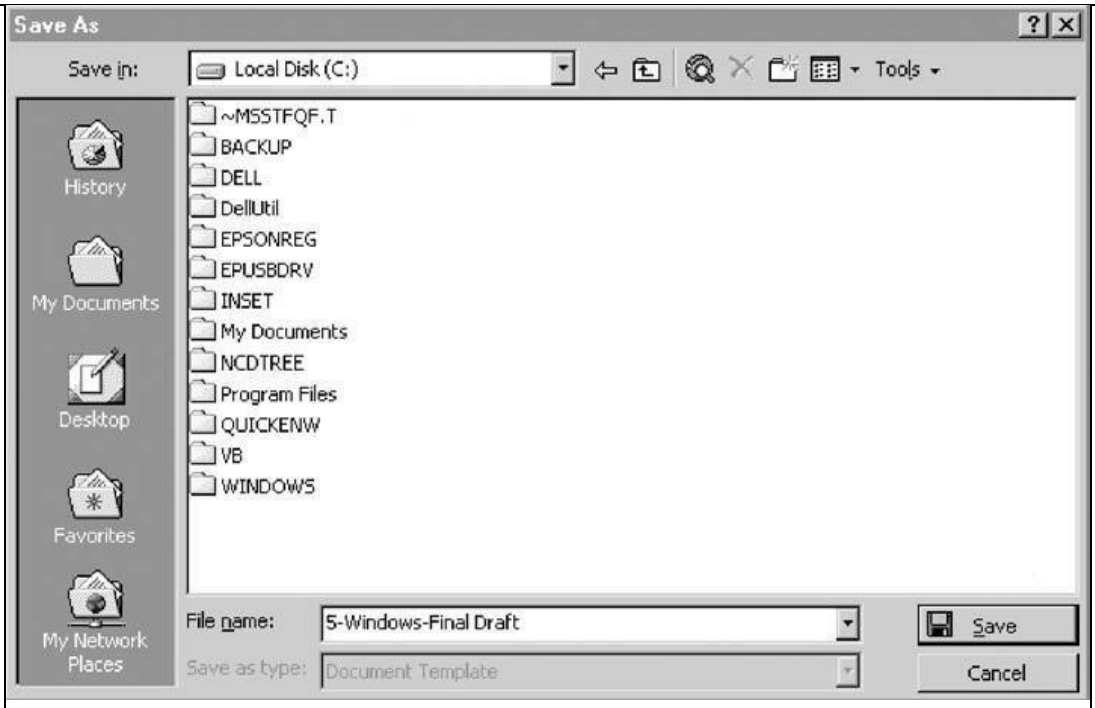
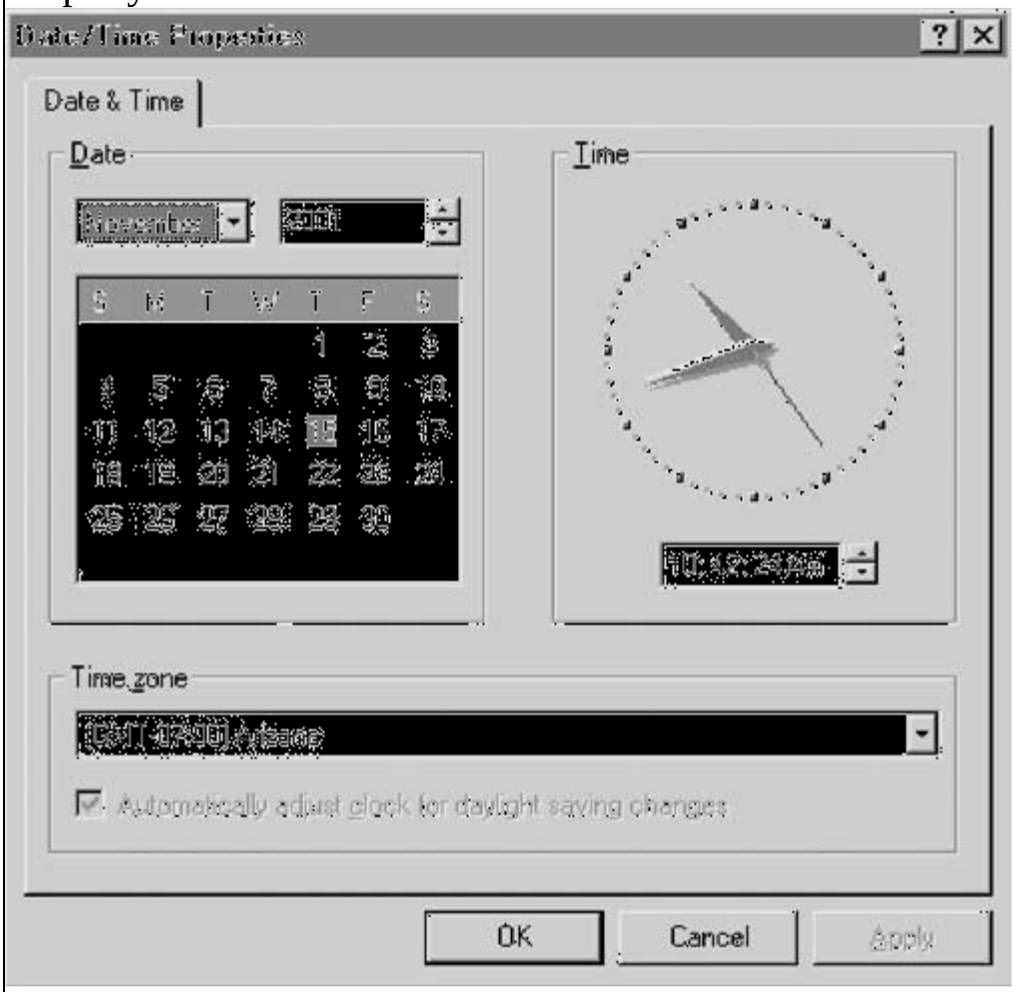



Figure 5.12 Unfolded Printers secondary window.

Dialog box



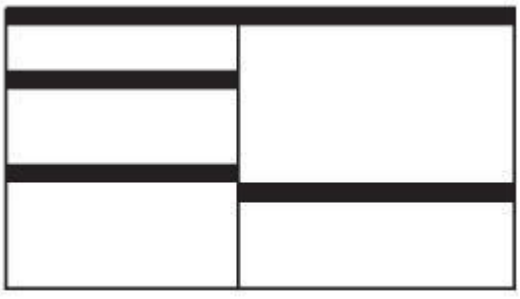

Property sheet




Message Boxes


Palette Windows


Pop-up Windows


| | | | | |
|---|---|---|---|---|
| 5(A) | **Give short notes on windows presentation styles and explain various window management techniques?**<br><br>The presentation style of a window refers to its spatial relationship to other windows. There are two basic styles, commonly called **tiled or overlapping.**<br>**I) Tiled Windows**<br><br>  <br>**Figure 5.4** Tiled windows.<br><br>Tiled windows, the first and oldest kind of window, are felt to have these **advantages**:<br>• The system usually allocates and positions windows for the user, eliminating the necessity to make positioning decisions. | [7] | CO2 | L2 |

- 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**

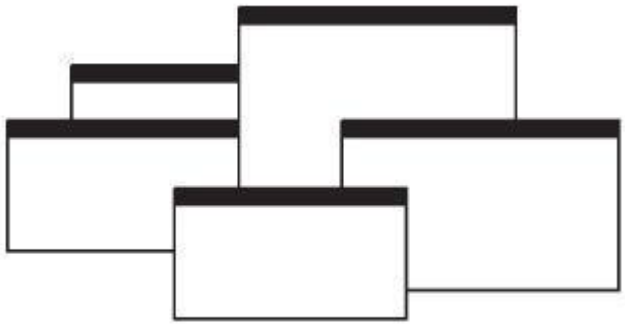


**Figure 5.5** Overlapping windows.

- 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**

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

**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.

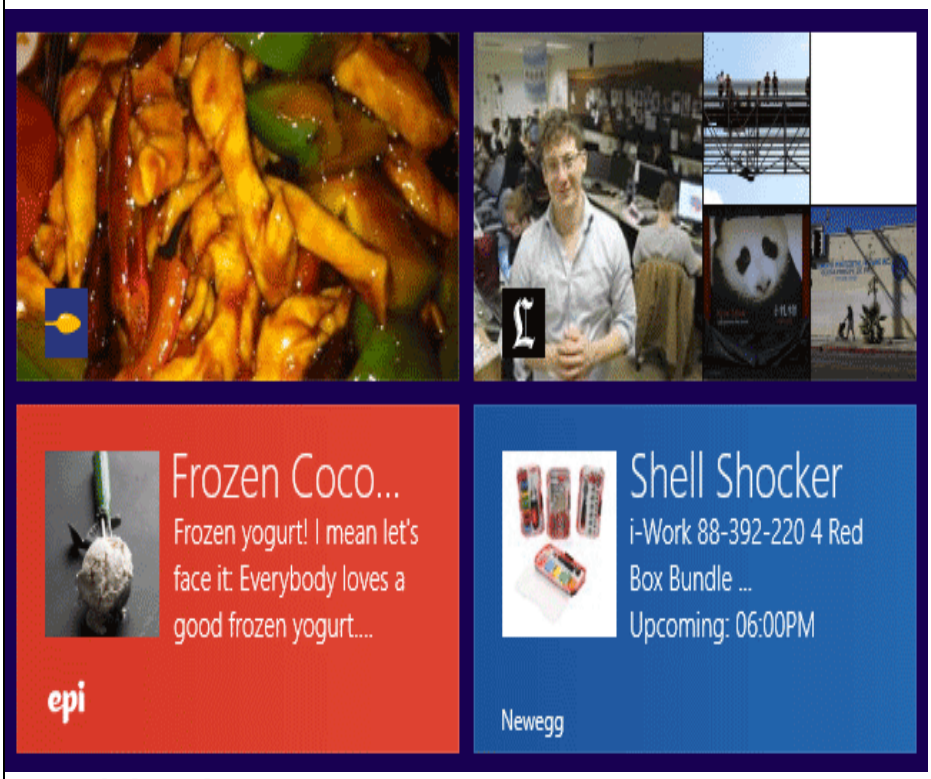
2. *A multiple-document interface*- A technique for managing a set of windows where documents are opened into windows.

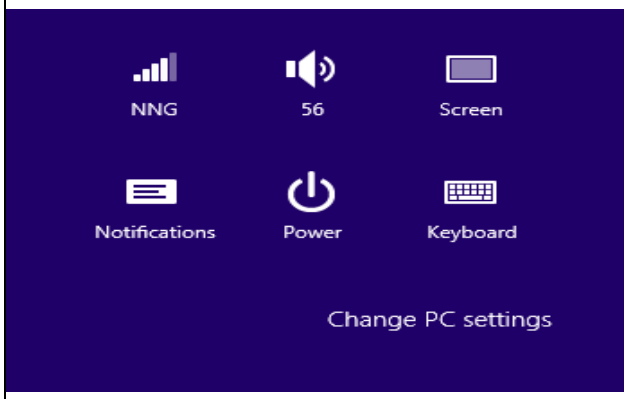
Contains:

• 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.<br>**4.** *projects*.-A technique that consists of a container: a project window holding a set of Objects | | | |
| 5(B) | **In the UI design shown below identify the window presentation style. Also explain how usability can be improved.**<br>  <br>It is a tiled window presentation<br><br>Though the tiles represent 4 different apps<br>Newegg is the only app that includes its full name in the tile.<br>Other 3 apps have no names, so the user cannot identify and use aa appropriate app. | [3] | CO1 | L2 |
| 6(A) | **Case study:**<br>**Social media and other easily accessible online distractions make it hard for us to stay focused on our tasks and make it difficult for us to do our work efficiently. If there is a simple App that lets us create a To-do list for all the tasks that has to be completed in day. What will be the simple UI design solution for the App .sketch down the design solution?**<br><br>Marks to be given<br>Considering if things like<br>• Calender<br>• To do list<br>• Task completed for the day<br>• Tasks yet to be completed are present in the design. | [6] | CO2 | L4 |
| 6(B) | **In the basic window design given below, what are your observations? List at least 5 flaws.** | [4] | CO1 | L2 |




**Observations**
Design for window OS
**Flaws**
Low information density
Change for settings is not a clickable command
Icons are flat and monochromatic
Poor distinctiveness of tab selection

**CO PO Mapping**

| Course Outcomes | | Modules covered | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 | PSO4 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO1 | Describe the characteristics of Graphics Interface and its principles. | **1 & 2** | 2 | 3 | - | 3 | - | 3 | - | - | - | - | - | - | - | - | 3 | 2 |
| CO2 | Analyze, design and evaluate user interface design | **1,2,3,4 & 5** | 2 | 3 | 3 | 3 | - | 3 | - | - | - | - | - | - | - | - | 3 | 2 |
| CO3 | Explain the components of web systems | **2,3 & 4** | 2 | 3 | 2 | 3 | - | 3 | - | - | - | - | - | - | - | - | 3 | 2 |
| CO4 | Demonstrate the guidelines of multimedia. | **2,3 & 4** | 2 | 3 | 2 | 3 | - | 3 | - | - | - | - | - | - | - | - | 3 | 2 |
| CO5 | Understand the prototype and kinds of test | **5** | 2 | 3 | 2 | 3 | - | 3 | - | - | - | - | - | - | - | - | 3 | 2 |

| COGNITIVE LEVEL | REVISED BLOOMS TAXONOMY KEYWORDS |
|---|---|
| L1 | List, define, tell, describe, identify, show, label, collect, examine, tabulate, quote, name, who, when, where, etc. |
| L2 | summarize, describe, interpret, contrast, predict, associate, distinguish, estimate, differentiate, discuss, extend |
| L3 | Apply, demonstrate, calculate, complete, illustrate, show, solve, examine, modify, relate, change, classify, experiment, discover. |
| L4 | Analyze, separate, order, explain, connect, classify, arrange, divide, compare, select, explain, infer. |
| L5 | Assess, decide, rank, grade, test, measure, recommend, convince, select, judge, explain, discriminate, support, conclude, compare, summarize. |

| PROGRAM OUTCOMES (PO), PROGRAM SPECIFIC OUTCOMES (PSO) | | | | CORRELATION LEVELS | |
|---|---|---|---|---|---|
| PO1 | Engineering knowledge | PO7 | Environment and sustainability | 0 | No Correlation |
| PO2 | Problem analysis | PO8 | Ethics | 1 | Slight/Low |
| PO3 | Design/development of solutions | PO9 | Individual and team work | 2 | Moderate/ Medium |
| PO4 | Conduct investigations of complex problems | PO10 | Communication | 3 | Substantial/ High |
| PO5 | Modern tool usage | PO11 | Project management and finance | | |
| PO6 | The Engineer and society | PO12 | Life-long learning | | |
| PSO1 | Develop applications using different stacks of web and programming technologies | | | | |
| PSO2 | Design and develop secure, parallel, distributed, networked, and digital systems | | | | |
| PSO3 | Apply software engineering methods to design, develop, test and manage software systems. | | | | |
| PSO4 | Develop intelligent applications for business and industry | | | | |