

Lesson 1 of 5: Text Editors

Outline

❖ Text Editors

- Overview of Editing process
- User interface
- Editor Structure

❖ Interactive Debugging systems

- Debugging functions and capabilities
- Relationships with Other parts of the system
- User Interface Criteria

Text editors

- ❖ An interactive editor is a computer program that allows a user to create and revise a target document.
- ❖ “Document” – objects such as computer programs, text, equations, tables, diagrams, line art and photographs – any thing that one might find on a printed page.
- ❖ Hence, a **text editor** is a type of program used for editing plain text files.

Overview of editing process

❖ Four task:

- **Select** the part of the target document to be viewed and manipulated.
- Determine how to **format** this view on-line and how to display it.
- Specify and execute operations that **modify** the target document.
- **Update** the view appropriately.

1.Selection

- ❖ Selection of a part of the document to be viewed & documented involves *traveling* through the document to locate the area of interest
- ❖ It is accomplished by *next screenful*, *bottom* and *find pattern*.
- ❖ Traveling specifies where the area of interest is whereas the selection of what is to be viewed and manipulated there is controlled by *filtering*
- ❖ *Filtering* extracts the relevant subset of the target document at the point of interest, such as next screenful of text or the next statement

2. Formatting

- ❖ Formatting determines how the result of filtering will be seen as a visible representation(the *view*) on the display screen or other device.

3. Actual editing

- ❖ The target document is created or altered with a set of operations such as *insert*, *delete*, *replace*, *move* and *copy*.
- ❖ Specialized to operate on *elements* meaningful according to the type of editors.
- ❖ For example, Elements in case of *manuscript-oriented editors* are single characters, words, lines, sentences, etc. whereas Elements in case of *program-oriented editor* are identifiers, keywords and statements.

4.Updating

- ❖ The view is properly updated and displayed on an output device

A Simple Scenario

- ❖ For deleting the first three words of a view:
 - The user travels to the end of the document.
 - A screenful of text would be filtered.
 - This segment of text is then formatted (deletion)
 - The view is displayed on an output device

User Interface

- ❖ **Conceptual Model**-Line and Screen editors
- ❖ **Input Devices**-Text, Button and Locator
- ❖ **Output Devices**-Teletypewriters, glass teletypes
- ❖ **Interaction Language**-Typed, Function Key & Menu orientations

Conceptual model

- ❖ Interactive editor is presented with a *conceptual model* of the editing system.
- ❖ The conceptual model is a abstract framework on which the editor & the world on which it operated are based
- ❖ In other words, it provides a clear abstraction of the target document and its elements with a set of guidelines describing the effects of operations on these elements
- ❖ Line Editor and Screen Editor

Conceptual Model -Line Editor

- ❖ Simulated the world of the Key punch (forefathers of Text Editors)
- ❖ Allowed operations on numbered sequence of 80-character card-image lines, either within a single line or on an integral number of lines.



Conceptual Model -Screen Editor

- ❖ Document is represented as quarter-plane of text lines, unbounded both down and to the right.
- ❖ Operations manipulate the portions of this quarter-plane without regard to line boundaries
- ❖ The user sees through the cutout, only a rectangular subset of this plane on a multi-line display terminal.
- ❖ The cutout can be moved to left or right, up or down, to display other portions of the document

Usage of a Screen Editor



Input Devices

- ❖ Input Devices are used to enter elements of the text being edited, to enter commands, and to designate editable elements
- ❖ Three categories-
 - Text/String devices
 - Button/Choice devices
 - Locator devices
 - Voice-Input devices (Future advancement)

Input Devices -Text/String devices

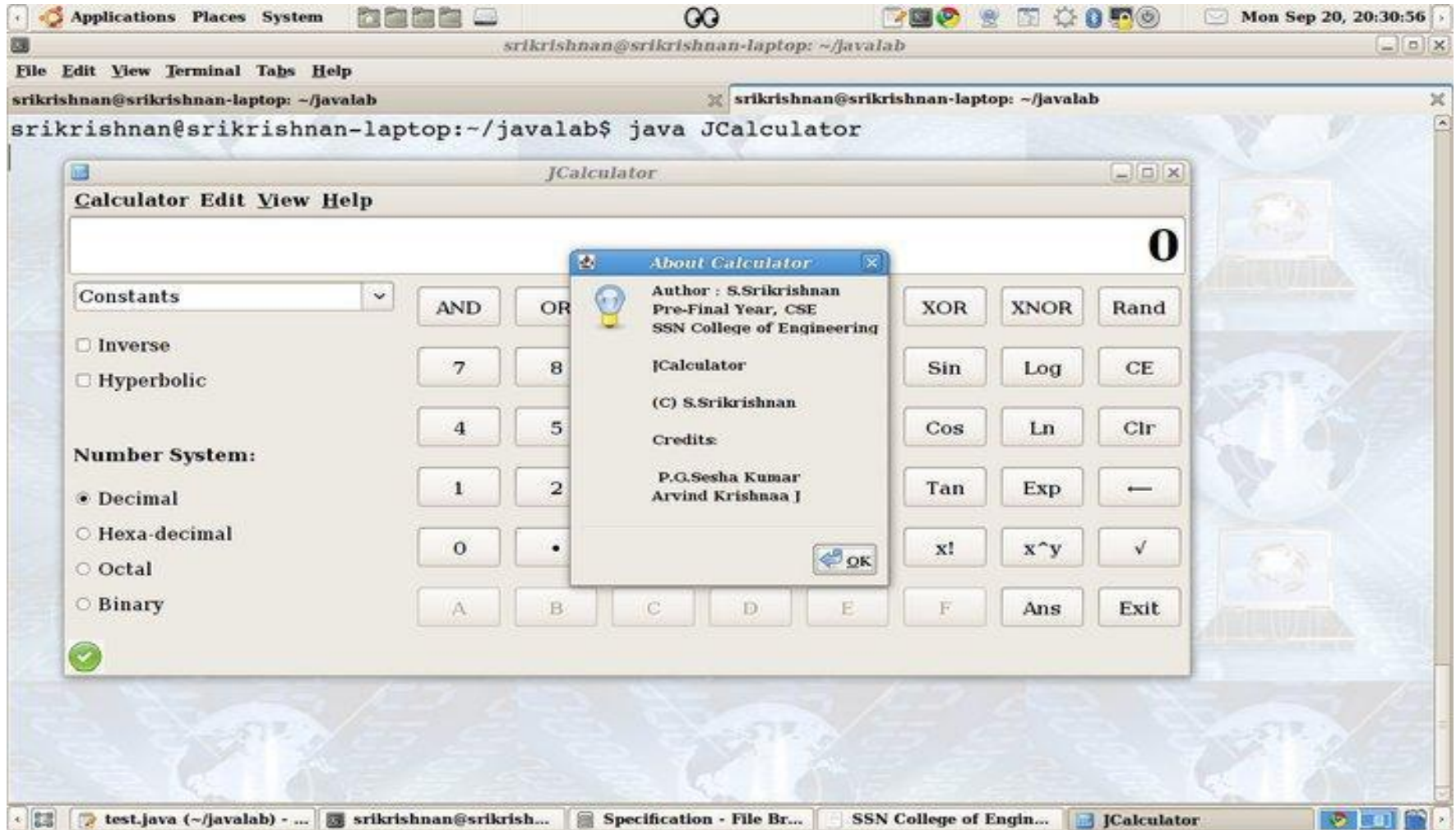
- ❖ Typically, typewriter like keyboards on which the user presses and release keys, sending a unique code for each key.
- ❖ Virtually all current computer keyboards are of QWERTY variety.
- ❖ Advantageous several alternative keyboard arrangements, though proposed, are hardly to be accepted due to the retraining effort



Input Devices -Button/Choice devices

- ❖ Generates an interrupt or system-flag, usually causing invocation of an associated application-program action.
- ❖ Includes a set of special function keys on an alphanumeric keyboard or on the display itself.
- ❖ Alternatively, buttons are simulated in software by displaying text strings or symbols on the screen
- ❖ The user chooses a string or symbol instead of pressing a button.

E.g. for Button/Choice Devices



Input Devices-Locator Devices

- ❖ Two dimensional analog-digital converters that position a cursor symbol on the screen by observing the user's movement of device.
- ❖ The most common devices are:
 - Mouse
 - Data Tablet
 - Arrow Keys

Data Tablet

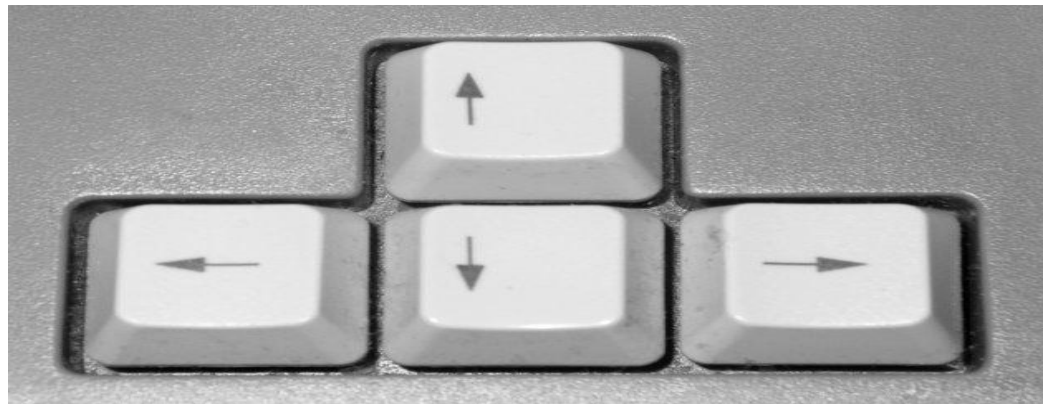
- ❖ Data Tablet is a flat, rectangular, electro-magnetically sensitive panel. A stylus is moved over the surface. (E.g. iPad™)
- ❖ Data Tablet returns the coordinates of the position of the stylus on the panel to the program.
- ❖ The program then maps these coordinates to the screen coordinates and move the cursor to corresponding screen position.

Data Tablet



Arrow Keys

- ❖ Simulate locator devices
- ❖ Up, down, left or right
- ❖ Arrow Keys generates an appropriate character sequence
- ❖ The program interprets this sequence and moves the cursor in the direction of the arrow on the key pressed



Input devices -Voice-Input devices

- ❖ Translates the spoken words to their textual equivalent.
- ❖ Currently available for command input in some systems(even in Windows™)
- ❖ It proves to be the text input devices of the future

Output devices

- ❖ Lets the user to view the elements being edited and the results of the editing operations
- ❖ The first output devices were ***teletypewriters*** and other ***character pointing terminals***.



- ❖ Next, ***glass teletypes*** based on cathode ray tube(CRT) technology used the CRT screen essentially to simulate a hard copy teletypewriter.



- ❖ Initial glass teletypes performed only few operations like backspacing elegantly.
- ❖ Today's advanced CRT terminals use hardware assistance for such features as moving cursor, inserting & deleting characters etc.
- ❖ Realistic facsimiles of hard-copy documents are produced by *professional workstations* based on high resolution display computers that supports multiple proportionally spaced character fonts

Interaction language

❖ Three types:

- Typing-oriented
- Function key-oriented
- Menu-oriented

Typing-Oriented

- ❖ Typing-Oriented or Text-Command oriented is the oldest of the major editor interfaces.
- ❖ The user communicates with the editor by typing text strings both for command names and for operands
- ❖ These strings are sent to the editor and are usually echoed to the output device.

Drawbacks in Typing-oriented

- ❖ Requires the user to remember the exact form of all the commands.
- ❖ The complex command language or less-frequently used commands require referring to a manual or an on-line help function.
- ❖ Typing required can be time consuming, especially for inexperienced users.
- ❖ Requires familiarity with the system and the languages with expertise in typing.

Function key-oriented

- ❖ The Function-key interface addresses the deficiencies of typing oriented systems.
- ❖ Each command has associated marked key on the user keyboard.
- ❖ E.g.: *the insert character* command is associated with it a key marked INSERT.

Function key-oriented

- ❖ Function key command specification is typically coupled with cursor-key movement for specifying operands, which eliminates much typing.
- ❖ For common commands, a single key is pressed.
- ❖ For less-frequently used commands, alternative textual syntax such as *special keys* or *overloaded keys* are used.

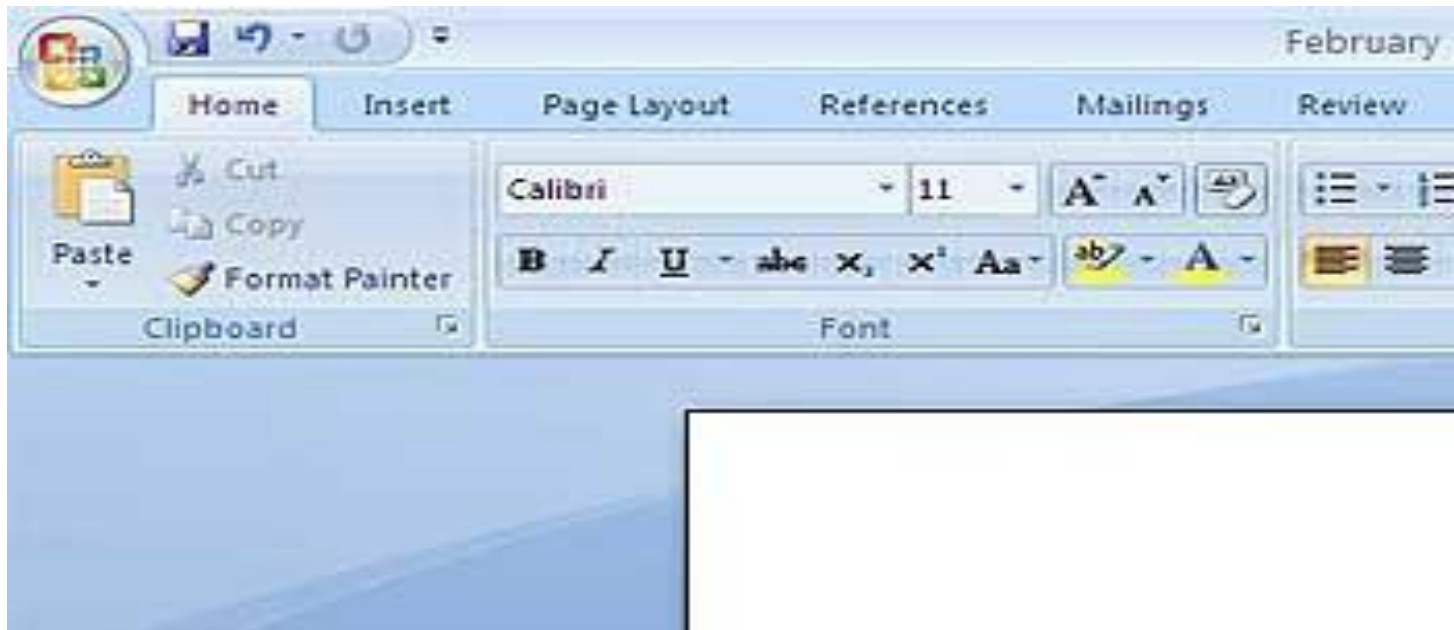


Function key-oriented

- ❖ *Special Keys*-Shift the standard function-key interpretations. E.g.: SHIFT key on a typewriter shifts from lowercase to uppercase
- ❖ *Overloaded Keys*-The standard alphanumeric keyboard is often overloaded to simulate function keys.
 - E.g.: Ctrl + N , Alt + F4 etc.
- ❖ *Disadvantages*: Functional key-oriented systems often have either too few keys, requiring multiple-keystroke commands or have many unique keys resulting in an unwieldy keyboard

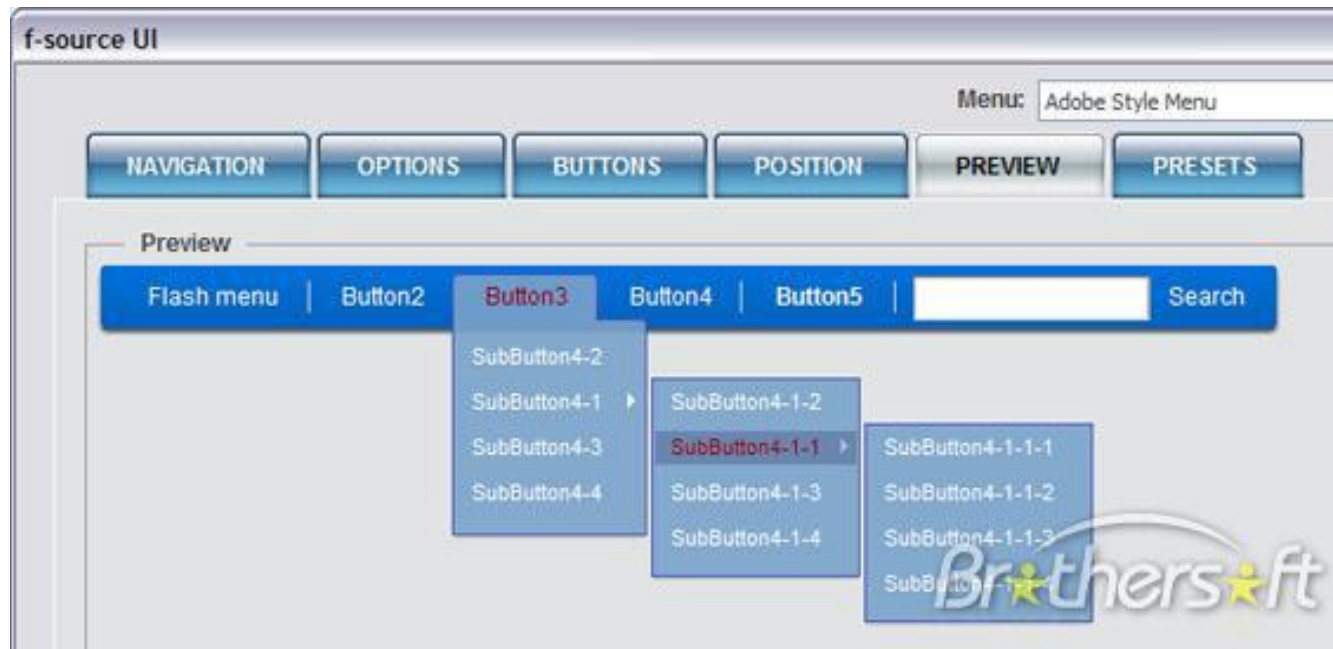
Menu-Oriented

- ❖ A *menu* is a multiple choice set of text strings or *icons*, which are graphic symbols that represent objects or operations. The user can perform action by selecting items from the menu.



Problems with Menu-oriented

- ❖ Many possible actions and several choices are required to complete an action.
- ❖ The display area for the menu is usually limited. Hence the user is presented with several consecutive methods in hierarchy before the appropriate command and its option appear



Overcoming the drawbacks

- ❖ Some Menu-Oriented programs allow to turn off menu control and return to a typing or function-key interfaces.
- ❖ Most-used functions on the main menu and the secondary menus to handle the less-frequently used functions.
- ❖ Displaying the menu only when user asks for it