AWT PROGRAMMING - 1
#LECTURE - 39

- applet for webs.
- java. applet. Applet.
- They can't access our own system & memory area.
- They can't use our API.
- They are heavy-weight.

AWT

- Abstract Windowing Toolkit
- This is lightweight.
- It is a ste of API for UI part of JDK

GUI with component

Component class

→ Button, checkbox, List, choic, frame, Panel, Lable, List, scrollbar, textarea, textfield.

METHOD

→ add (componen c)
→ Set Size (int w, int h)
→ SetLayout (Layoutmanage m)
→ SetVisiblity (boolean status)

Lightweight process

FRAME (creating a frame)

→ It is a top level window with title & border.
→ Subclass of constructor

"Public class Frame extends window implement MenuContainer."

Constructor ⇒ frame ()
frame (string)

```
Frame f = new Frame ("frame in Java");
f.resize (500,500);
f.set background (Color.blue);
f.show ();
```

PANEL (creating a panel)

→ It is the simplest container class.
→ Provides space where application can attach other components.
→ Inherits container

"public class Panel extends container implement accesible."

CONST ⇒ Panel()
Panel (LayoutManger Layout)

addNotify(), getAccessibleContext() } methods.

```
Panel p = new Panel();
f.add(P);
f.show();
```

Button

input focus □
onclick ■

it u want action performed it should implement actionListner + new Listener various event.

Button()
Button(string)

extend fram

```
Button b = new Button("click");
b Set Boun(100, 100, 100, 100); → x, y, width, ht.
set Size(300,300)
setLayout(null);
setvisible(true);
```

CHECKBOX

checkbox c = new checkbox(" ");

LABEL

Component for placing text.
Single line of read-only text.

Label()
Label (string, allignment)

TEXTFIELD

Textfield()
Textfield(int) → col count.
Textfield (string) → inbuil text.

TextArea

Display texts
Textarea() , (r, c) , (s, r, c) , (str, r, c, scrollbar)

List

scrolling of list of text items.

List()
List(row)
List(int r, boolean mode)

l.add()

CHOICE

c.add();

SCROLLBAR

scrollbar()
scrollbar(int orientation)
scrollbar(int ori, int val, int min, max) → initial val.

AWT PROGRAMMING-II
#LECTURE-40

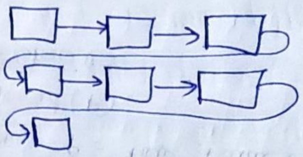
LAYOUT MANAGERS

each contaner has an layout manager.
↳ layout manager interface.

→ Flowlayout
→ BorderLayout
→ GridLayout
→ CardLayout

FLOWLAYOUT MANAGER

It is a default layout.
It is used to arrange components in a line.



hgap- hori gap
vgap- veri gap

FlowLayout()
FlowLayout(Int align) horizontal /vertical
FlowLayout(int align, int hgap, int vgap)

f. SetLayout(new FlowLayout(FlowLayout.RIGHT));

METHOD ⇒ center, leading, left, right, trialing.

BORDERLAYOUT MANAGER

East, west, North, south & centre.
BorderLayout()
JBorderLayout(hgap, vgap)
FLow Layout(align, hgap, vgap)

CENET, EAST, NORTH, SOUTH, WEST

f.add(b5, BorderLayout. NORTH);
f.add(b1, BorderLayout. EAST);

GRID LAYOUT

→ arrange components in rectangular grid
→ one component is displayed in rectangle

```
GridLayout ()
GridLayout (r, c)
Gridlayout (r,c, hgap, vgap);

f. setLayout (new GridLayou (3,3));
        ~ FLowLayout...
```

CardLayout

only one component is visible at a time on clicking each time card moves.

```
cardlayout()
CardLayout (hgap, Vgap).
```

↓ Contain parent

```
next( )
prev ( )
first()
Last ()
Show(contain parent, string Name)

Public boolean keyDown (Event e, int Key){
        Layout.next(this)
        return true;
}
```

EVENT HANDLING

→ To add life to the component.
→ Event-driven programming

Mouse event → moving, dragging, entering, exiting
Keyboard event → pressing & releasing
windows event → destroying, iconifying, exposing & deiconifying
Scrolling event → scrolling line, page.