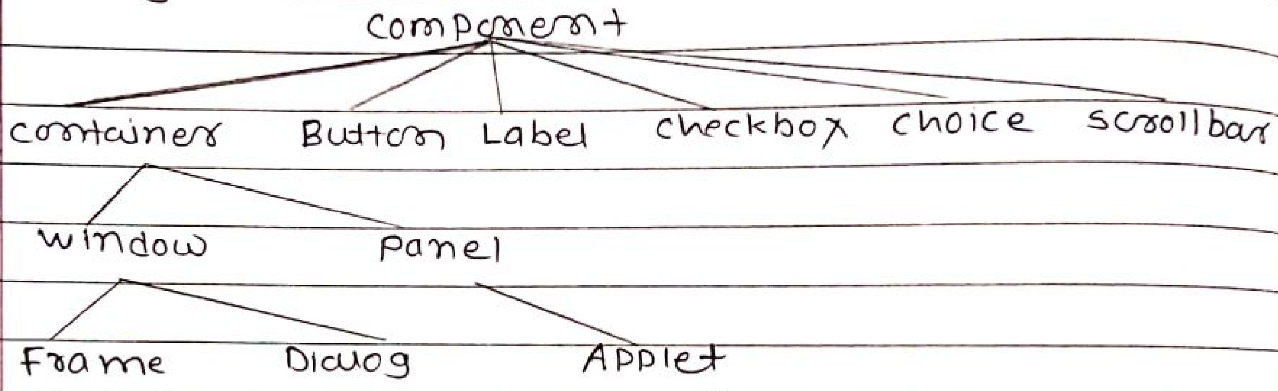
Unit-1 AWT (Notes)

① AWT stands for Abstract Window Toolkit.
② AWT contains large number of built-in classes, interfaces and methods which allows creating and managing GUI applications.

③ package : java.awt



(class Hierarchy in awt package)

④ component class
It is abstract superclass for all AWT components/ controls.

* Methods
① add (component c)
② remove (component c)
③ remove All ()
④ setLayout (Layout Manager m)
⑤ setVisible (Boolean value)

⑤ Container
ⓘ container which will holds another components
ⓘ container is a subclass of component class
ⓘⓘ It has two subclasses window and panel

⑥ Window

(i) The window class creates a top-level window.

(ii) It is baseclass for Frame and Dialog.

* Frame :- It is a subclass of window and has a title bar, menu bar, borders, and resizing corners.

* Canvas

Canvas is a blank rectangular area where the user can draw or trap input from the user

⑦ Panel

(i) A panel is a window that does not contain a title bar, menu bar, or border.

(ii) Applet is subclass of Panel.

⑧ Applet

package: java.applet

Life cycle methods of Applet:

```
① public void init() {}
② public void paint(Graphics g) {}
③ public void start () {}
④ public void stop () {}
```

⑨ Frame

package: java.awt

* Constructor

1. Frame()
2. Frame(String title)

Note:- By default Frame is hidden and its size is 0px.

* Methods
1. void setTitle(String title)
2. void setVisible(true)
3. void setSize(int width, int height)
4. void setSize(Dimension d)

* Creating Frame window:-
There are two methods:
1. By extending Frame class (Inheritance)
2. By creating Frame object (Association.)

* By Extending Frame class
```
import java.awt.*;
class class_name extends Frame
{
 class_name()
{ //body of constructor
 }
public static void main(String args[])
{ //body of main()
 }
}
```

* By using Frame class object (Association)
```
import java.awt.*;
Class class_name
{ class_name()
 { Frame f= new Frame();
  f.setVisible(true);
 }
public static void main(String args[])
 { // body of main() }
 }
```

⑩ Features of AWT component
① platform dependent
② Heavy weighted.

⑪ Steps to add AWT component in a container
1. Declare object of component
e.g. Button b;
2. Initialize this object of component using constructor
e.g. b= new Button("submit");
3. Insert component into container using add()
e.g. add (b)

AWT Components / controls

1. Label
① It is a passive component. (user cannot edit text)
⑪ used to display single line of read only text.

* Constructor
① Label()
② Label(String str)
③ Label(String str, int align)

* Methods
① void setText(String str)
② String getText()
③ void setAlignment (int how)
④ int getAlignment()

2. Button

① Button is a control component that has a label and generates an event when pressed.

* Constructor

① Button()

② Button(String str)

* Methods

① void setLabel(String str)

② String getLabel()

3. TextField

① TextField is a single line area used to take user input. allows to editing single line text.

* ② By default TextField size is 5px.

③ TexComponent is Super Class of Text Field.

* Constructor

① TextField()

② TextField(int width)

③ TextField(String Str)

④ TextField(String str, int width)

* Methods

① void setText(String str)

② String getText()

③ setEchoChar(char ch)

④ void setEditable(boolean value)

4. Checkbox

① A checkbox is a control that is used to turn an option on or off.

② Every checkbox has label and it selects / deselects multiple options.

* Constructor

① Checkbox()

② Checkbox(String str)

③ Checkbox(String str, boolean state)

* Methods

① void setLabel(String str)

② String getLabel()

③ void setState(boolean state)

④ boolean getState()

* How to create Radiobutton in AWT?

① CheckboxGroup class is used to make group of Check boxes.

⑪ CheckboxGroup object is used to make radio button.

* Constructor of CheckboxGroup Class
  CheckboxGroup()

* Methods of CheckboxGroup Class:
  Checkbox get ~~Sted~~ectedCheckbox()

* To create radiobutton following constructors of Checkbox class are used.

① Checkbox(String str, CheckboxGroup obj, boolean state)

② Checkbox(String str, boolean state, CheckboxGroup obj)

5. List

① The object of List class represents a list of text items.

⑪ with the help of the List class, user can choose either one or multiple items.

* constructor

① List()

② List (int num rows)

③ List(int num_rows , boolean mode)

* Methods

① void add(String item)

② void add (String item, int index)

③ String getItem(int index)

④ int getItemCount()

⑤ int getSelectedIndex()

⑥ String getSelectedItem()

⑦ void removeAll (int position)

⑧ void removeAll()

6. Choice

① The object of choice class is used to show popup menu of choices.

⑪ Choice / Item selected by user is show on the top of a menu.

* constructor

Choice ()

* Methods:

1. void add(String item)
2. String getItem(int index)
3. int getItemCount()
4. int getSelectedIndex()
5. String getSelectedItem()
6. void insert(String item, int index)
7. void remove(int position)
8. void removeAll()

7. Text Area

i) It is multiline area used to take user input in GUI Application.

ii) TextComponent is superclass of TEXT Area Class

* Constructor

1. TextArea()
2. TextArea(int Lines, int numChars)
3. TextArea(String str)
4. TextArea(String str, int numLines, int numChars)

* Methods:

1. String getText()
2. void setEditable(Boolean value)
3. boolean isEditable()
4. void append(String str)
5. void insert(String str, int index)
6. void setText(String str)

8. Scroollbar

* Constructor

① Scroollbar(): creates vertical scroollbar.

② Scroollbar(int style):- creates scroll bar with given style:

Scroollbar. HORIZONTAL

Scroollbar. VERTICAL

* Difference between List & Choice.

	List	Choice
①	A list may be displayed in such a way that several list items are visible.	A choice is displayed in compact from that requires you to pull it down to see list of availabel choice.
②	A List supports the selection of one or more list items.	only one item selected from a choice.
③	A list is any enumeration of set items	Choice is the act of picking or deciding between two more possibilities.

* Difference between Radio button & checkbox

	Radio button	Checkbox
①	Used only when one option is selected.	Used checkbox to allows one or many options.

②	It is a single control unit.	It is multiple control unit.
③	Radio button represented as a small circle.	Checkbox represented a small square.
④	Radio button have only two states true & false	Checkbox have 3 states checked, unchecked and Indeterminate.

* Difference between TextArea & TextField.

	TextArea	TextField
①	TextArea component can allow users to enter multiline of text.	TextField component can allow users to enter single line of text.
②	Syntax:- TextArea t=new TextArea();	Syntax:- TextField tf=new TextField();
③	It is multiple line textbox with width & height & ~~[illegible]~~ horizontal scrollbar.	In a text Field it is a single line texbox
④	Class: TextArea	Class: TextField

Scanned with CamScanner

⑥ Layout Manager.
Layout manager automatically arranges awt controls within a window using setLayout().

* Syntax :- void setLayout(LayoutManager m)

* We can manually arrange components in container using :
  setBounds(int x, int y, int width, int height)

* Types of LayoutManager.
1. FlowLayout.
2. BorderLayout.
3. Grid Layout.
4. GridBagLayout.
5. CardLayout.

1. FlowLayout
(i) FlowLayout arranges component one after another starting from top-left corner, left to right and top to bottom.
(ii) A small space by default is 5px is left between each component.
* (iii) FlowLayout is a default layout manage for Applet.

* Constructor
① FlowLayout()
② FlowLayout(int align)
③ FlowLayout(int align, int hgap, int vgap)

* Constant Defined by FlowLayout:
1. FlowLayout.LEFT
2. FlowLayout.RIGHT
3. FlowLayout.CENTER

2. BorderLayout
(i) The BorderLayout is used to arrange the component in five regions: north, south, east, west and center.
(ii) Each region (area) may contain one component only.
* (iii) BorderLayout is the default layout of Frame or window.

* Constructor.
(1) BorderLayout()
(2) BorderLayout(int hgap, int vgap)

Note* When we use BorderLayout for container then the add() method changes as.
add(Component obj, BorderLayout region)

* Constant Defined by BorderLayout:
1. BorderLayout.NORTH
2. BorderLayout.SOUTH
3. BorderLayout.EAST
4. BorderLayout.WEST
5. BorderLayout.CENTER.

3. GridLayout

(i) The GridLayout is used to arrange the components in rows and columns / rectangular grid.

(ii) one component is placed in each cell of grid.

* Constructor:

(1) GridLayout()
Creates grid of single row and column equal to number of components.

(2) GridLayout(int row, int col)
creates grid with given number of row and col.

(3) GridLayout(int row, int col, int hgap, int vgap)
creates grid with given number of row and col. component will have specified horizontal and vertical space between them.

4. GridBagLayout

(i) A GridBagLayout places components in a grid of rows and columns, allowing specified components to span multiple rows or colums.

(ii) Not all rows necessarilly have the same height

(iii) GridBagLayout components are associated with the instance of GridBagConstraints. These constraints are used to define the components display area and their positions.

Scanned with CamScanner

* Constructors:
Grid Bag Layout ():
It is used to creates grid bag Layout managers.

* Method:
add ((Component obj, Grid Bag Constraints gbc)

* int gridx
Specifies the cell x co-ordinate in component's display area, where the first cell in a row has gridx=0

* int gridy
specifies the cell y co-ordinate of the component's display area, where the topmost cell has gridy=0

* int gridwidth
specifies the number of cells in arow for the component's display area.

* Int grid height
specifies the number of cells in a column for the components display area.

* insets(int top, int bottom, int left, int right)
This field specifies the minimum amount of space between the component and the edges of its display area.

* Menu & MenuBar

* ① MenuBar is associated with top-level window only i.e. only Frame can have MenuBar.

* ⑪ In java MenuBar is created by coupling 3 different classes: MenuBar, Menu, MenuItem.

$$\begin{tikzcd}
 & \text{MenuComponent} \arrow[dl] \arrow[dr] & & \\
\text{MenuBar} & & \text{MenuItem} \arrow[dl] \arrow[dr] & \\
 & \text{Menu} & & \text{CheckboxMenuItem}
\end{tikzcd}$$

1. MenuBar

① MenuBar is created using object of class MenuBar

* Constructer : MenuBar()

⑪ Inserting Menubar in frame window using set MenuBar(MenuBar m)

* Methods of MenuBar class:

① void add(Menu obj)

② int getMenuCount()

③ void remove(int index)

2. Menu

① The object of Menu class is a pull down menu component which is displayed on the menu bar. It Inherits the MenuItem class.

* Constructer: ⓘ Menu()

⑪ Menu (String menu_name)

⑪⑪ Menu (String menu-name, boolean remove)

3. MenuItem

① The object of MenuItem class adds a simple labeled menu item on menu.

* Constructor:

① MenuItem()

② MenuItem(String title)

③ MenuItem(String title, MenuShortcut key)

* Methods:

① setEnabled(boolean value)

② setLabel(String title)

4. CheckboxMenuItem()

① The object of CheckboxMenuItem class creates menuitem that toggles each time on clicking.

* Constructors:

① CheckboxMenuItem()

② CheckboxMenuItem(String title)

③ CheckboxMenuItem(String title, boolean state)

* Methods:

① int getState()

② void setState(Boolean state)