

Accredited by NBA - AICTE and Accredited by NAAC - UGC with 'A++' Grade

Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

Department of Computer Applications

Course Code: 23CAT606

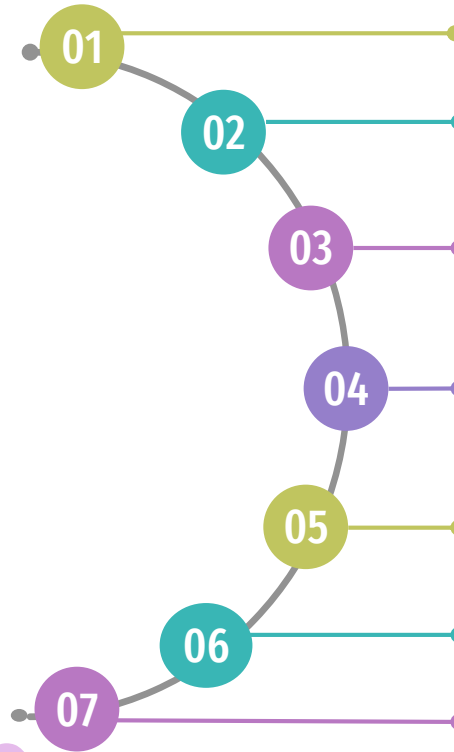
Course Name: Java Programming

Unit II: Package

Topic 10: AWT



Recall-Applet



Applet

Life Cycle

Graphics Class

Predefined Methods

Two ways to run Applet

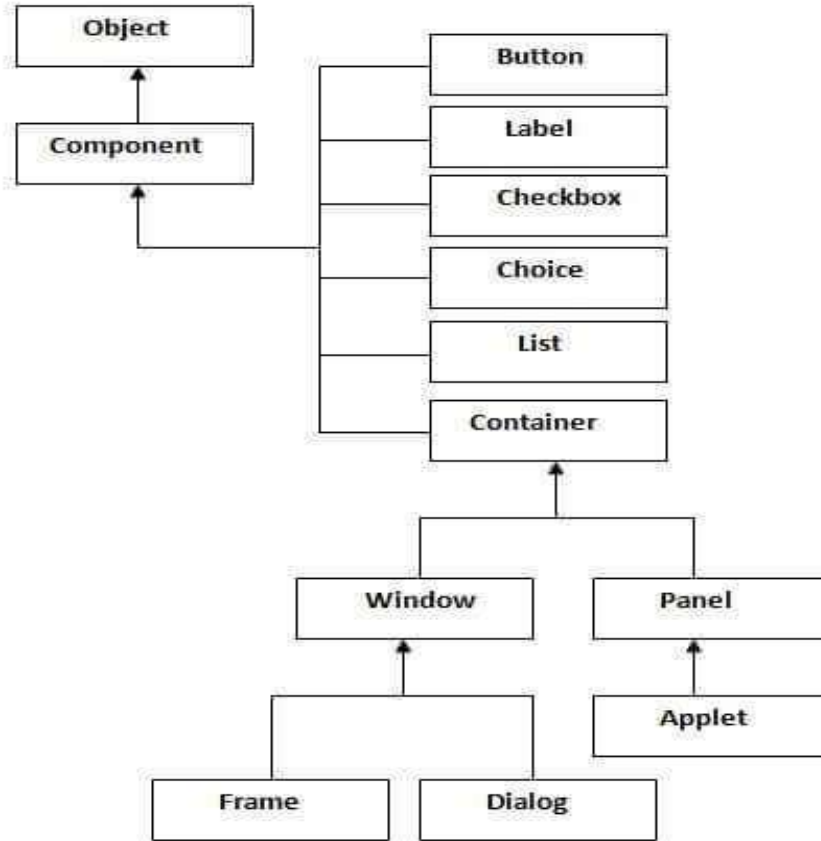
Example Programs

Assessment



1. **Java AWT (Abstract Window Toolkit)** is an **API** to **develop GUI** or **window-based** applications in java.
2. Java AWT components are platform-dependent i.e. components are displayed according to the view of operating system. AWT is heavyweight i.e. its components are using the resources of OS.
3. The java.awt package provides classes for AWT api such as TextField, Label, TextArea, RadioButton, CheckBox, Choice, List etc.

AWT Hierarchy



USING AWT COMPONENTS



- COMPONENT

- CANVAS
- SCROLLBAR
- BUTTON
- CHECKBOX
- LABEL
- LIST
- CHOICE
- TEXT
 - TEXTAREA
 - TEXTFIELD

- COMPONENT

- CONTAINER
 - PANEL
 - WINDOW
 - DIALOG
 - FRAME
- MENU COMPONENT
 - MENU ITEM
 - MENU



Container

The Container is a component in AWT that can contain other components like buttons, textfields, labels etc. The classes that extends Container class are known as container such as **Frame, Dialog and Panel**.

Window

The window is the container that have no borders and menu bars. Use frame, dialog or another window for creating a window.

Panel

The Panel is the container that doesn't contain title bar and menu bars. It can have other components like button, textfield etc.

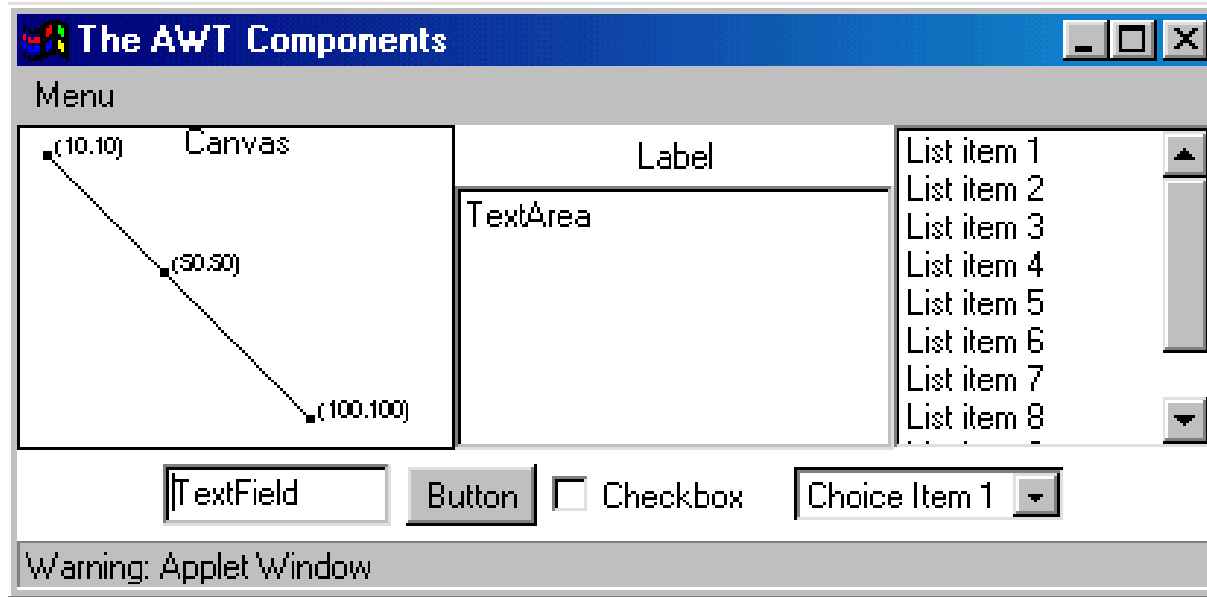
Frame

The Frame is the container that contain title bar and can have menu bars. It can have other components like button, textfield etc.



Method	Description
<code>public void add(Component c)</code>	inserts a component on this component.
<code>public void setSize(int width,int height)</code>	sets the size (width and height) of the component.
<code>public void setLayout(LayoutManager m)</code>	defines the layout manager for the component.
<code>public void setVisible(boolean status)</code>	changes the visibility of the component, by default false.

AWT COMPONENTS



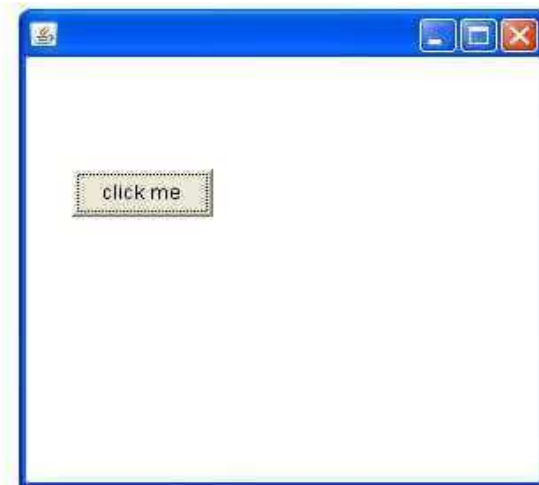


```
import java.awt.*;

public class TestFrame extends Frame {
    public TestFrame(String title){
        super(title);
    }
    public static void main(String[] args){
        Frame f = new TestFrame("The AWT Components");
        f.setSize(400,400);
        f.setLocation(100,100);
        f.show();
    }
}
```

HOW TO USE BUTTONS?

```
import java.awt.*;
class First2
{
First2()
{
    Frame f=new Frame();
    Button b=new Button("click me");
    b.setBounds(30,50,80,30);
    f.add(b);
    f.setSize(300,300);
    f.setLayout(null);
    f.setVisible(true);
}
public static void main(String args[]){
    First2 f=new First2();
    }}
}
```



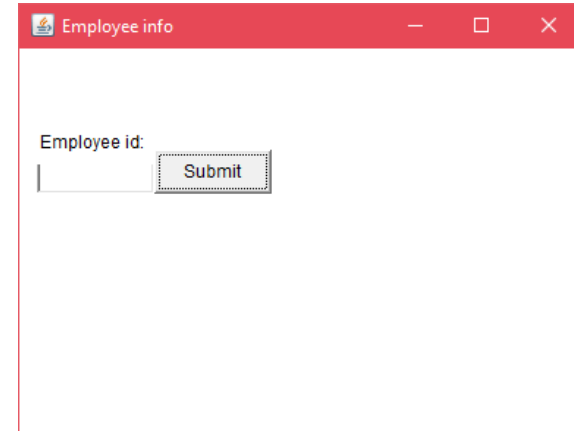
HOW TO USE LABELS, TEXTFIELD?



```
import java.awt.*;

class AWTEExample2 {
    AWTEExample2() {
        Frame f = new Frame();
        Label l = new Label("Employee id:");
        Button b = new Button("Submit");
        TextField t = new TextField();
        l.setBounds(20, 80, 80, 30); t.setBounds(20, 100, 80, 30);
        b.setBounds(100, 100, 80, 30);
        f.add(b); f.add(l); f.add(t);
        f.setSize(400,300); f.setTitle("Employee info");
        f.setLayout(null); f.setVisible(true);
    }
}
```

```
public static void main(String args[]) {
    AWTEExample2 awt_obj = new AWTEExample2();
}
```



HOW TO USE CHECKBOXES?

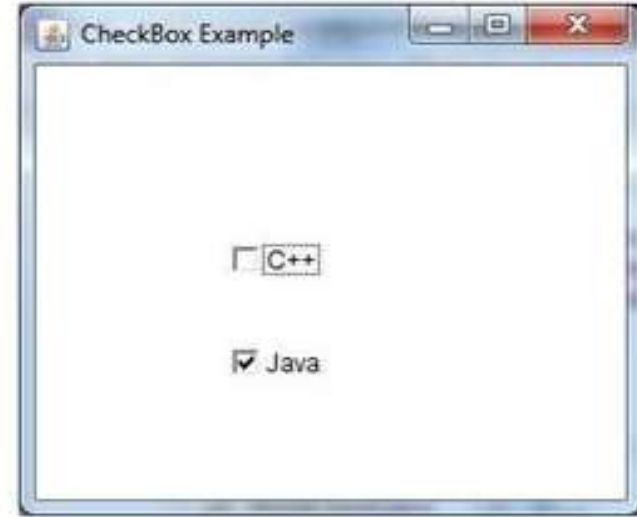


```
import java.awt.*;

class AWTEExample2 {
    AWTEExample2() {
        Frame f = new Frame("Checkbox Example");
        Checkbox checkbox1=new Checkbox("C++");
        checkbox1.setBounds(100,100,50,50);

        Checkbox checkbox2=new Checkbox("Java",true);
checkbox1.setBounds(100,150,50,50);
        f.add(checkbox1);
        f.add(checkbox2);
        f.setSize(400,400);
        f.setLayout(null);
        f.setVisible(true);
    }
}
```

Output:



HOW TO USE CHOICES?



```
public class TestChoice extends Frame {
    public TestChoice(String title){
        super(title);
```

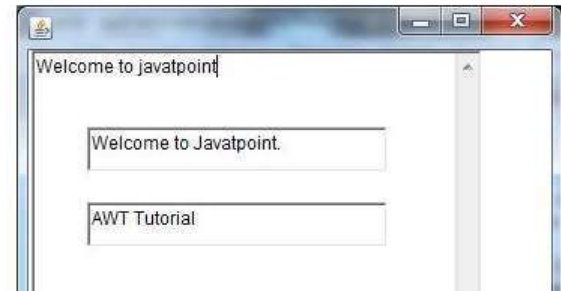
```
        Choice choice = new Choice();
        choice.add("Item 1");
        choice.add("Item 2");
        choice.add("Item 3");
        choice.add("Item 4");
        choice.add("Item 5");
        add(choice);
    }
```



HOW TO USE TEXTFIELD & TEXTAREA

```
public class TestText extends Frame{
    public TestText(){
        TextField textField = new TextField(20);
        TextArea textArea = new TextArea(5, 20);
        textField.setText("Welcome to Javatpoint");
        textArea.setText("AWT Tutorial");
        add(textField,"North");
        add(textArea,"South");
    }
    ...
}
```

Output:



HOW TO USE LISTS?



```
public class TestList extends Frame {
    public TestList(){
        List l = new List(5, true);
        l.add("Item 1");
        l.add("Item 2");
        l.add("Item 3");
        l.add("Item 4");
        l.add("Item 5");
    }
}
```

Output:



HOW TO USE MENUS?



```
import java.awt.*;
```

```
class MenuExample
```

```
{
```

```
    MenuExample(){
```

```
        Frame f= new Frame("Menu and MenuItem Example");
```

```
        MenuBar mb=new MenuBar();
```

```
        Menu menu=new Menu("Menu");
```

```
        Menu submenu=new Menu("Sub Menu");
```

```
        MenuItem i1=new MenuItem("Item 1");
```

```
        MenuItem i2=new MenuItem("Item 2");
```

```
        MenuItem i3=new MenuItem("Item 3");
```

```
        MenuItem i4=new MenuItem("Item 4");
```

```
        MenuItem i5=new MenuItem("Item 5");
```

```
        menu.add(i1);
```

```
        menu.add(i2);
```

```
        menu.add(i3);
```

```
        submenu.add(i4);
```

```
        submenu.add(i5);
```

```
        menu.add(submenu);
```

```
        mb.add(menu);
```

```
        f.setMenuBar(mb);
```

```
        f.setSize(400,400);
```

```
        f.setLayout(null);
```

```
        f.setVisible(true);
```

```
    }
```

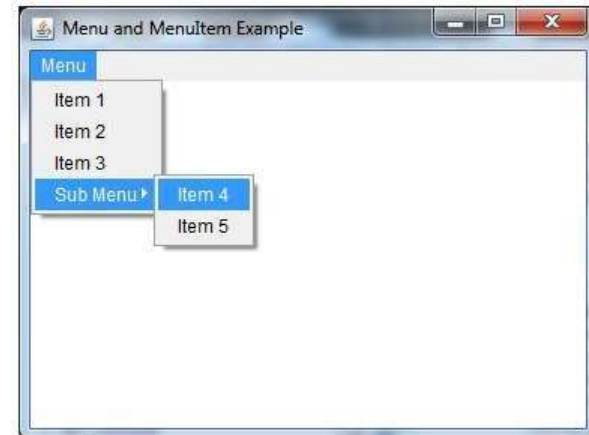
```
    public static void main(String args[])
```

```
    {
```

```
        new MenuExample();
```

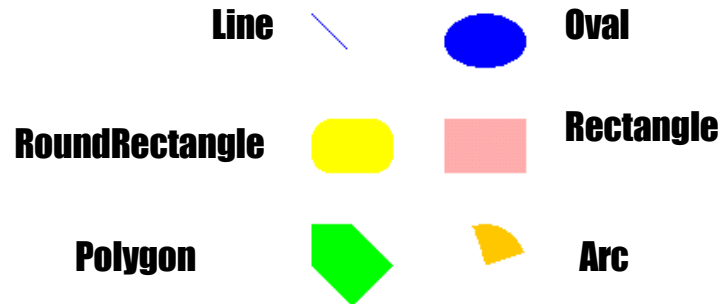
```
    }
```

```
}
```



HOW TO USE CANVASES AND GRAPHICS PRIMITIVES?

- FOR DRAWING GEOMETRIC SHAPES, TEXTS, AND IMAGES
- AN ABSTRACT CLASS
 - THE EXTENDED CLASS MUST OVERRIDE **PAINT()**



DRAWLINE(X1,Y1,X2,Y2)

```
class MyCanvas extends Canvas
{
public void paint(Graphics g)
{
g.setColor(Color.blue);
    int x1 = 161,
        y1 = 186,
        x2 = 181,
        y2 = 206;
g.drawLine(x1,y1,x2,y2); }
```

(161,186)



(181,206)

FILLOVAL(X,Y,W,H) DRAWOVAL(X,Y,W,H)

```
g.setColor(Color.blue);  
{  
    int x = 239,  
        y = 186,  
        w = 48,  
        h = 32;  
    g.fillOval(x,y,w,h);  
}
```



FILLPOLYGON(INT[] XS, INT[] YS) DRAWPOLYGON(INT[] XS, INT[] YS)

```

g.setColor(Color.green);
{
    int xpoints[] = {25, 145, 25, 145, 25};
    int ypoints[] = {25, 25, 145, 145, 25};
    int npoints = 5;
    g.fillPolygon(xpoints, ypoints, npoints);
}
    
```



FILLRECT(X, Y, W, H) DRAWRECT(X, Y, W, H)

```
g.setColor(Color.pink);  
{  
  int x = 239,  
    y = 248,  
    w = 48,  
    h = 32;  
  g.fillRect(x,y,w,h);  
}
```



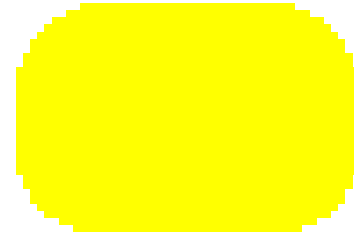
FILLROUNDRECT(X, Y, W, H, RW, RH)

DRAWROUNDRECT(X, Y, W, H, RW, RH)

```

g.setColor(Color.yellow);
{
  int x = 161,
    y = 248,
    w = 48,
    h = 32,
    roundW = 25,
    roundH = 25;
  g.fillRoundRect(x,y,w,h,roundW,roundH);
}

```



FILLARC(X, Y, W, H, SA, A)

DRAWARC(X, Y, W, H, SA, A)

```

g.setColor(Color.orange);
{
  int x = 239,
    y = 310,
    w = 48,
    h = 48,
    startAngle = 20,
    angle = 90;
  g.fillArc(x,y,w,h,startAngle,angle);
}

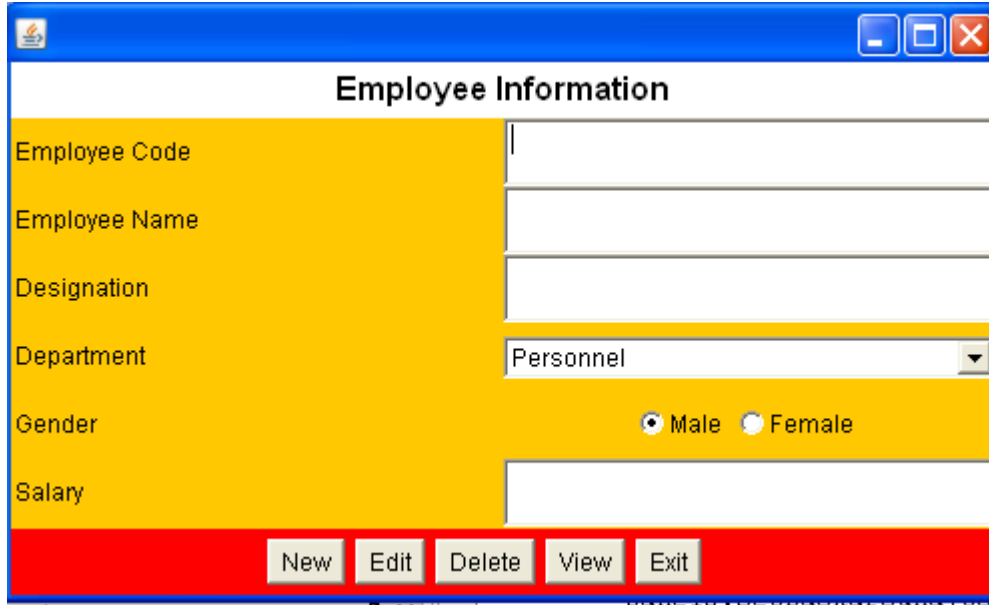
```



DRAWSTRING, FONT, & FONTMETRICS

```
class MyCanvas extends Canvas {  
  
    public void paint(Graphics g)  
    {  
        g.setFont(new Font("Dialog",0,20));  
        FontMetrics fm = g.getFontMetrics();  
        int x = 100;  
        int y = 100;  
        g.drawString("Hello",x,y);  
        y = y+fm.getHeight();  
        g.drawString("World",x,y);  
    }  
}
```


GUI-Employee Management System



The screenshot shows a window titled "Employee Information" with a yellow background. It contains several input fields: "Employee Code", "Employee Name", "Designation", "Department" (a dropdown menu currently showing "Personnel"), "Gender" (radio buttons for "Male" and "Female", with "Male" selected), and "Salary". At the bottom, there is a red bar with five buttons: "New", "Edit", "Delete", "View", and "Exit".

Summary

- 1
- 2
- 3



Reference

1. Herbert Schildt “ The Complete Reference Java 2, 8th edition , Tata McGraw Hill, 2011
2. Ralph Bravaco, Shai Simonson, “Java Programming: From the Ground up Tata McGraw Hill, 2012
3. <https://www.javatpoint.com>

*Thank
you*

