

Essentials of Information Technology

PC-CS-305

GUI using AWT & Swing

Objectives



- Describe the AWT package and its components.
- Define the terms *containers*, *components*, and *layout managers*, and how they work together to build a graphical user interface (GUI).

gauravgambhir.cse@piet.co.in

Objectives



- Add components to a container
- Use the Frame and Panel containers appropriately

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

Overview

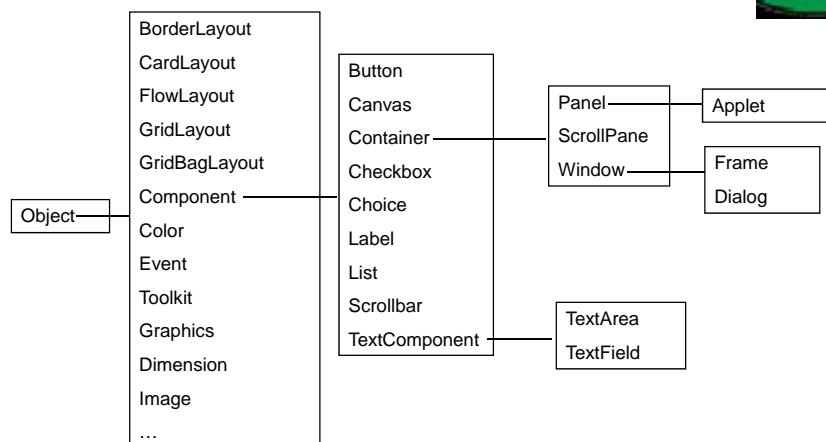


- A typical Graphical User Interface (GUI) uses several graphical components
 - Common examples:
 - Windows, buttons, menus, text fields, check boxes, images
- Java provides a basic graphical toolkit called AWT (Abstract Windowing Toolkit)
- Java 2 provides more advanced graphical capability via Swing
 - Swing is a part of JFC (Java Foundation Classes)

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

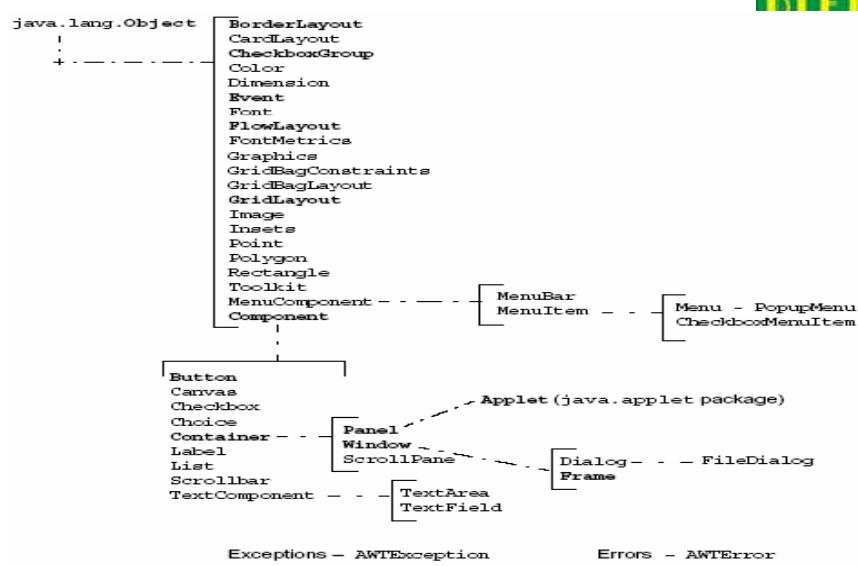
The java.awt package



gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

The java.awt Package



PC-CS-305: Essentials of Information Technology

Abstract Window Toolkit



- A rich set of tools provided by Java to help the programmer build Graphical User Interfaces (GUI) with ease.
- GUIs built with AWT promises that:
 - The final interface can run anywhere!
 - This is unlike many GUI applications which are mainly hardware or operating system dependant.

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

Abstract Window Toolkit



- Provides graphical user interface (GUI) components that are used in all Java applets and application.
- Contains classes that can be extended and their properties inherited; classes can also be abstract.
- Ensures that every GUI component that is displayed on the screen is a subclass of the abstract class Component or MenuComponent

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

Abstract Window Toolkit



- Has Container, which is an abstract subclass of Component and includes two subclasses:
 - Panel
 - Window

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

Containers

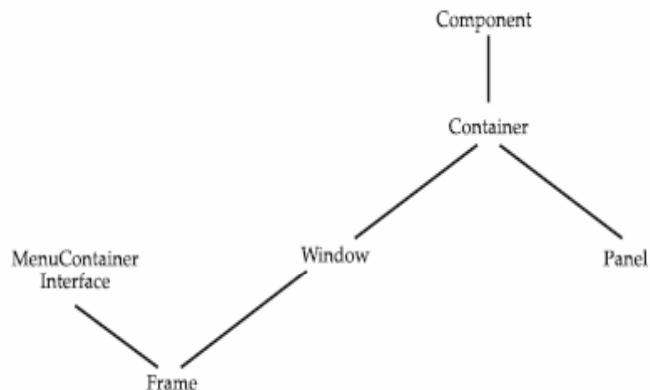


- The two main types of containers are Window and Panel
- A Window is a free floating window on the display
- A Panel is a container of GUI components that must exist in the context of some other container, such as a window or applet

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

The java.awt Package



gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

Building GUI



The position and size of a component in a container is determined by a layout manager.

You can control the size or position of components by disabling the layout manager.

You must then use `setLocation()`, `setSize()`, or `setBounds()` on components to locate them in the container.

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

AWT Basics - Frames



- A **Frame** is a top-level window with a title and a border. In the Java API, its object hierarchy looks something like this;

```
java.lang.Object
  |
  +--java.awt.Component
  |
  +--java.awt.Container
  |
  +--java.awt.Window
  |
  +--java.awt.Frame
```

- As you can see, the **Frame** inherits from a long line of superclasses, the immediate one being the **Window** class

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

Frames



- Are a subclass of Window
- Have title and resizing corners
- Are initially invisible, use setVisible(true) to expose the frame
- Have BorderLayout as the default layout manager
- Use the setLayout method to change the default layout manager

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

AWT Basics - Frames

Creating your first frame



```
import java.awt.*;
//Ch19_01_MyFrame
public class MyFrame extends Frame {

    public static void main(String[] args) {
        MyFrame me = new MyFrame();
    }
    public MyFrame() {
        setSize(150,150);
        setLocation(100,100);
        setTitle("My First java.awt.Frame!");
        setLayout(new FlowLayout());
        setVisible(true);
    }
}
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

AWT Basics - Frames

- `setSize()` - sets the size of the frame.
- `setLocation()` - sets the location of the frame within the desktop.
- `setTitle()` - sets the title of the frame.
- `setVisible()` - makes the frame visible to the user.



PC-CS-305: Essentials of Information Technology

Example

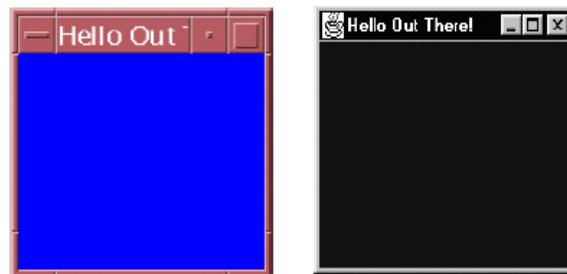


```
import java.awt.*;
public class FrameExample {
private Frame f;
public FrameExample(){
f = new Frame("Hello Out There!");
}
public void launchFrame() {
f.setSize(170,170);
f.setBackground(Color.blue);
f.setVisible(true);
}
public static void main(String args[]){
FrameExample guiWindow = new FrameExample();
guiWindow.launchFrame();
}
}
```

see *FrameExample.java*
gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

Output



gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

Panels



- Provide a space for components.
- Allow subpanels to have their own layout manager.

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

Example



```
import java.awt.*;
public class FrameWithPanel {
    private Frame f;Panel pan;
    public FrameWithPanel(String title){
        f = new Frame(title);
        pan = new Panel();
        public void launchFrame(){
            f.setSize(200,200);
            f.setBackground(Color.blue);
            f.setLayout(null); // Override default layout mgr
            pan.setSize(100,100);
            pan.setBackground(Color.yellow);
            f.add(pan);
            f.setVisible(true);
        }
    }
}
```

see *FrameWithPanel.java*
gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

Example

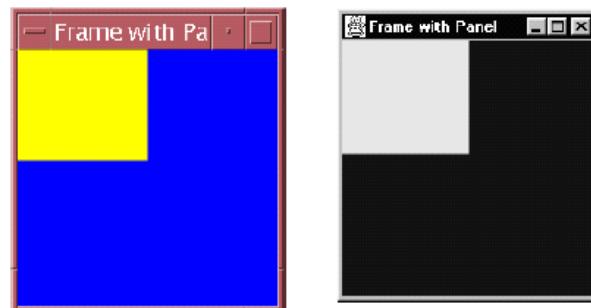


```
public static void main(String args[]){
    FrameWithPanel guiWindow =
        new FrameWithPanel("Frame with Panel");
    guiWindow.launchFrame();
}
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

Output



gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

AWT Basics - Label



- A **Label** object is a component for placing text in a container. It is used for displaying a single line of read-only text.
- The **text can be changed by the application**, but a user cannot edit it directly.
- The **Label** object is located in the following API hierarchy;

```
java.lang.Object
 |
 +--java.awt.Component
 |
 +--java.awt.Label
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

AWT Basics - Label



Creating your first Label

```
import java.awt.*;
//Ch19_02_MyFrame
public class MyFrame2 extends Frame {
    public static void main(String[] args) {
        MyFrame2 me = new MyFrame2();
    }
    public MyFrame2() {
        setSize(150,150);
        setLocation(100,100);
        setTitle("My First java.awt.Frame!");
        setLayout(new FlowLayout());
        Label l = new Label("java.awt.Label");
        add(l);
        setVisible(true);
    }
}
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

AWT Basics - Label



- The output of the program might look something like this :



gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

Labels



The easiest control to use is a label. A *label* is an object of type

Label, and it contains a string, which it displays.

Example LabelDemo.java

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

AWT Basics - TextField



- A **TextField** object is a text component that allows for the editing of a single line of text.
- This allows for **general user-input**.
- The **TextField** object is located in the following API hierarchy;

```
java.lang.Object  
|  
+--java.awt.Component  
|  
+--java.awt.TextField
```

gauravgambhir.cse@piet.co.in

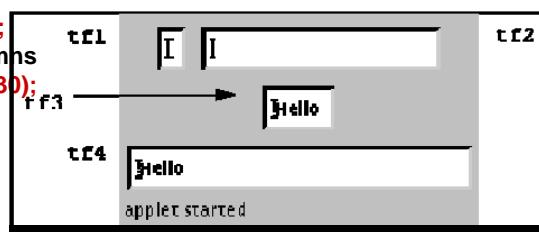
PC-CS-305: Essentials of Information Technology

AWT Basics - TextField



Various Variations of TextField

```
TextField tf1, tf2, tf3, tf4;  
// a blank text field  
tf1 = new TextField();  
// blank field of 20 columns  
tf2 = new TextField("", 20);  
// predefined text displayed  
tf3 = new TextField("Hello!");  
// predefined text in 30 columns  
tf4 = new TextField("Hello", 30);
```



gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

AWT Basics - TextField



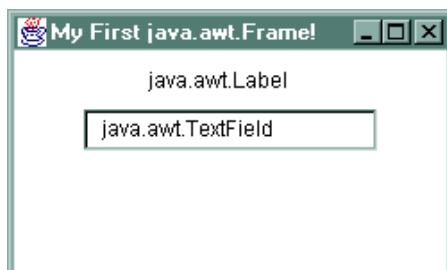
```
import java.awt.*;
//Ch19_03_MyFrame
public class MyFrame extends Frame {
    public static void main(String[] args) {
        MyFrame me = new MyFrame();
    }
    public MyFrame() {
        setSize(150,150);
        setLocation(100,100);
        setTitle("My First java.awt.Frame!");
        setLayout(new FlowLayout());
        Label l = new Label("java.awt.Label");
        add(l);
        TextField tf = new TextField("java.awt.TextField", 20);
        add(tf);
        setVisible(true);
    }
}
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

AWT Basics - TextField

- The output of the program might look something like this :



gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

AWT Basics - TextArea



- A **TextArea** object is a *multi-line region that displays text*.
- It can be set to allow editing or to be read-only.
- The **TextArea** object is located in the following API hierarchy;

```
java.lang.Object
 |
 +--java.awt.Component
   |
   +--java.awt.TextArea
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

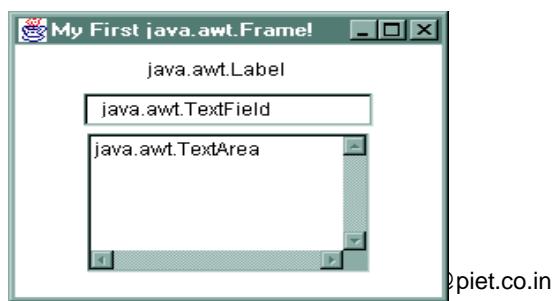
AWT Basics - TextArea

Creating a TextArea



```
TextArea ta = new TextArea("java.awt.TextArea", 5, 20);
add(ta);
```

By adding the codes above to the previous program, the output of the program might look something like this :



PC-CS-305: Essentials of Information Technology

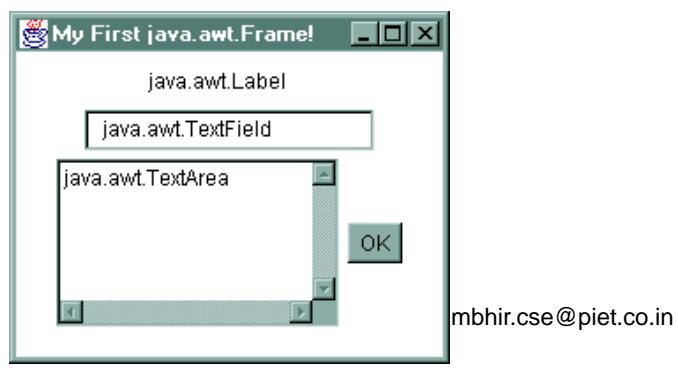
AWT Basics - Button

Creating a Button



```
Button b = new Button("OK");
add(b);
```

By adding the codes above to the previous program, the output of the program might look something like this :



PC-CS-305: Essentials of Information Technology

Buttons



A *push button* is a component that contains a label and that generates an event when it is pressed.

Example ButtonDemo.java

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

Buttons



Each time a button is pressed, an action event is generated.

This is sent to any listeners that previously registered an interest in receiving action event notifications from that component.

Each listener implements the **ActionListener** interface.

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

Buttons



Interface defines the **actionPerformed()** method, which is called when an event occurs.

An **ActionEvent** object is supplied as the argument to this method. It contains both a reference to the button that generated the event and a reference to the string that is the label of the button.

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

AWT Basics - List



- The **List** component presents the user with a scrolling list of text items
- The **List** can be set up so that the user can choose either **one item or multiple items.**
- The **List** object is located in the following API hierarchy;

```
java.lang.Object  
|  
+--java.awt.Component  
|  
+--java.awt.List
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

AWT Basics - List

Creating a List



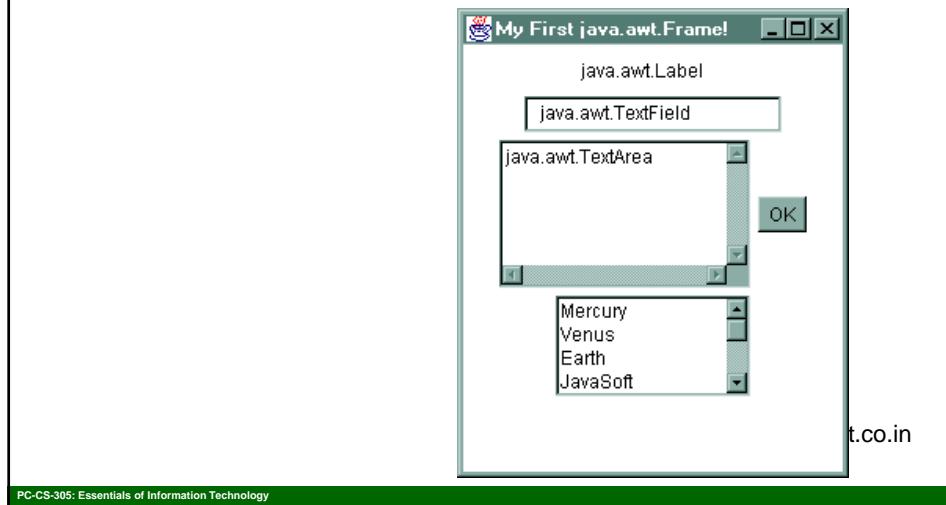
```
List lst = new List(4, false);  
  
lst.add("Mercury");  
lst.add("Venus");  
lst.add("Earth");  
lst.add("JavaSoft");  
lst.add("Mars");  
lst.add("Jupiter");  
lst.add("Saturn");  
lst.add("Uranus");  
lst.add("Neptune");  
lst.add("Pluto");  
  
add(lst);
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

AWT Basics - List

By adding the codes above to the previous program, the output of the program might look something like this :



List



The **List** class provides a compact, multiple-choice, scrolling selection list. Unlike the **Choice** object, which shows only the single selected item in the menu, a **List** object can be constructed to show any number of choices in the visible window.

Example ListDemo.java

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

AWT Basics - Choice



- The **Choice** class presents a pop-up menu of choices.
- The **Choice** object is located in the following API hierarchy;

```
java.lang.Object
 |
 +--java.awt.Component
   |
   +--java.awt.Choice
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

AWT Basics - Choice



Creating a Choice

```
Choice c = new Choice();
c.add("red");
c.add("green");
c.add("blue");
add(c);
```

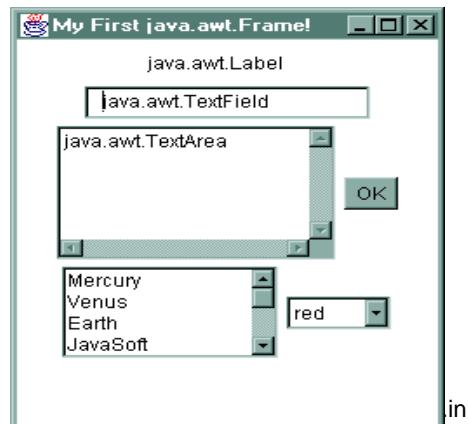
gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

AWT Basics - Choice



By adding the codes above to the previous program, the output of the program might look something like this :



PC-CS-305: Essentials of Information Technology

Choice List



The **Choice** class is used to create a *pop-up list* of items from which the user may choose. Thus, a **Choice** control is a form of menu. When inactive, a **Choice** component takes up only enough space to show the currently selected item.

Example ChoiceDemo.java

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

Check Boxes



A *check box* is a control that is used to turn an option on or off. It consists of a small box that can either contain a check mark or not.

Example CheckboxDemo.java

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

Using Java Swing



- Simply import the Swing package into a class that needs to use it
 - `import javax.swing.*;`
- Java Swing uses some classes from the AWT, so import that, too, when needed
 - `import java.awt.*;`
- Swing and AWT classes make great use of OO concepts
 - Abstraction and information-hiding
 - Inheritance and polymorphism

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

The JFrame Class



- Inherits more than 200 functions from other classes
 - In the awt package:
Frame → Window → Container → Component →
 - In the lang package
Object class
- Consult the documentation!
 - <http://java.sun.com/j2se/1.5.0/docs/api/>

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

A JFrame Example



```
import javax.swing.*;
public class JFrameExample extends JFrame
{
    public JFrameExample()
    {
        setTitle("This is a JFrame");
        setBounds(0, 0, 200, 200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }
    public static void main(String args[])
    {
        new JFrameExample();
    }
}
```



gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

Textual Components in Swing



- Java Swing provides four main textual component classes
 - `JLabel`, `JTextfield`, `JTextArea` and `JTextPane`
 - We shall look at the first three
- A `JLabel` component displays text
- A `JTextField` component permits users to enter a single line of text
- A `JTextArea` component permits users to enter many lines of text

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

JLabel



- To create a `JLabel` object:
 - `JLabel myLabel;`
`myLabel = new JLabel("Some text");`
- Some of the functions that can be called:
 - `setText(String s)`
 - `getText()`
 - `setFont(Font f)`

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

JTextField



- To create a JTextField object:
 - JTextField shapeDiameter;
shapeDiameter = new JTextField();
- Some of the functions that can be called:
 - setText(String s)
 - getText()
 - setEditable(boolean b)
 - setFont(Font f)

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

JTextArea



- To create a JTextArea object:
 - JTextArea displayArea;
displayArea = new JTextArea(15, 30);
- Some of the functions that can be called:
 - setText(String s)
 - append(String s)
 - getText()
 - setEditable(boolean b)
 - setFont(Font f)

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

JComboBox



- To create a JComboBox object:
 - ```
 JComboBox myCombo;
myCombo = new JComboBox();
```
  - Some of the functions that can be called:
    - `getItemAt(int index)`
    - `addItem(E item)`
    - `getItemCount()`
- Example `ComboBoxExample.java`

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## JTabbedPane



- To create a JTabbedPane object:
    - ```
JTabbedPane myPane;
myPane = new JTabbedPane();
```
 - Some of the functions that can be called:
 - `add(String title, Component component)`
 - `getTabCount()`
 - `indexOfTab(String title)`
- Example `TabbedPaneExample.java`

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

JRadioButton



- To create a JRadioButton object:
 - ```
RadioButton myButton;
myButton = new JRadioButton("male");
```
- Most of the functions that can be called are inherited from
  - javax.swing.AbstractButton class
  - javax.swing.JComponent class
  - java.awt.Container class
  - java.awt.Component class
  - java.lang.Object class
- Example RadioButtonExample.java

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Scrolling the Text Area



- If we append too many history entries, we have another problem
  - If the text area is not big enough, we lose the text
- The solution is to use a JScrollPane with the JTextArea

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## JScrollPane



- To create a `JScrollPane` object:
  - `JScrollPane scrollPane;  
scrollPane = new JScrollPane(displayArea);`
- Can also set the vertical and horizontal scrollbar policies:
  - `new JScrollPane(displayArea,  
JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,  
JScrollPane.HORIZONTAL_SCROLLBAR_NEVER);`
- Add the scroll pane instead of the text area to the frame
- Example : `JScrollPaneExample.java`

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## JTree



- To create a `JTree` object:
  - `JTree myTree;  
myTree = new JTree();`
- Some of the functions that can be called:
  - `addSelectionRow(int row)`
  - `cancelEditing()`
  - `getRowCount()`
- Example `TreeExample.java`

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## JTable



- To create a JTable object:
  - ```
• JTable myTable;
    myTable = new JTable(int numRows, int
    numColumns);
```
- Some of the functions that can be called:
 - `getRowCount()`
 - `getSelectedRow()`
 - `getTableHeader()`
- Example `TableExample.java`

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

Summary of Main Teaching Points



- We understood how to create and use
 - Frame, Panel, Label, TextField, TextArea , Button , List , Choice , CheckBox
 - JFrame, JLabel, JTextField , JTextArea , JScrollPane, JComboBox, JTabbedPane, JRadioButton, JTree, JTable

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

Question and Answer Session



Q & A

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology