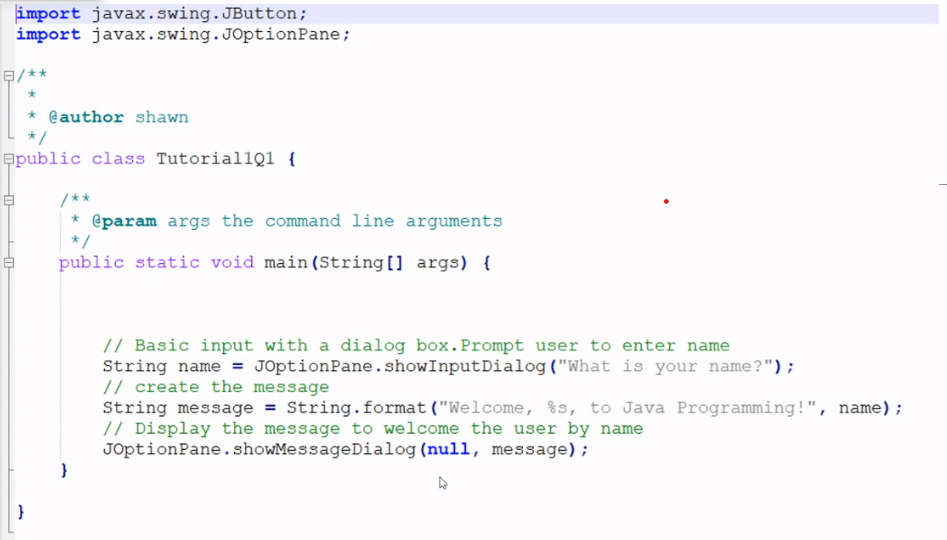
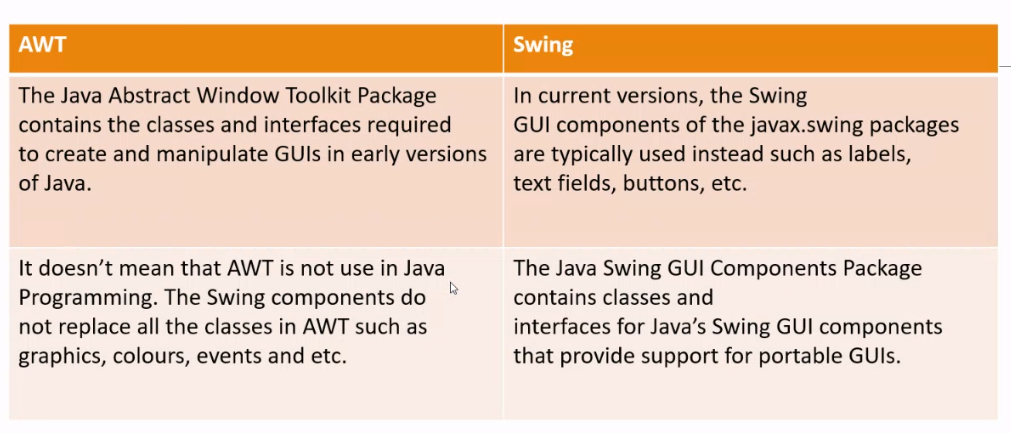
Tutorial 1

1. **ShowNameMsg Class**

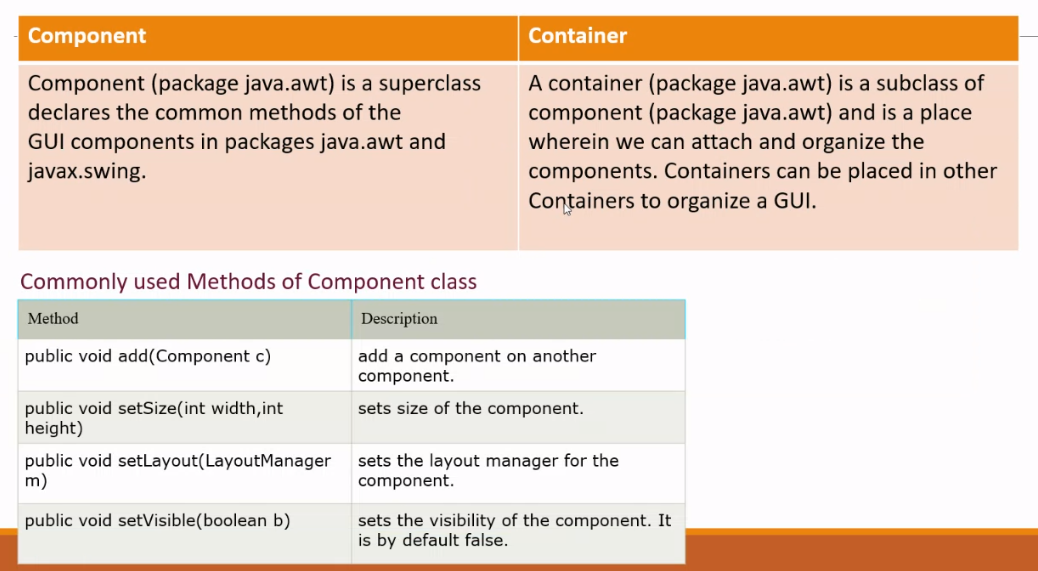
| import javax.swing.JOptionPane;  public class ShowNameMsg {  public static void main(String[] args){  String nameInput = JOptionPane.showInputDialog  ("What is your name : ");    JOptionPane.showMessageDialog(null,  "Welcome," + nameInput + ",to Java Programming"  );  }  } |
| --- |

Sample Answer :

1. Sample Answer:



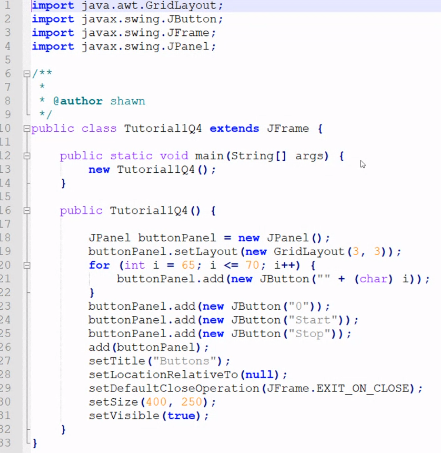
1. Sample Answer:



1. **ButtonsClass Class**

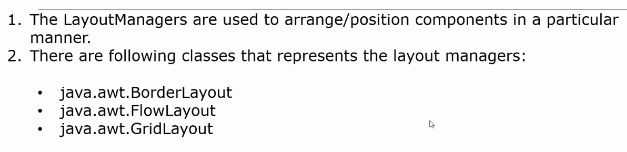
| import java.awt.\*;  import javax.swing.\*;  public class ButtonsClass extends JFrame {  ButtonsClass() {  JPanel buttonPanel = new JPanel();  buttonPanel.setLayout(new GridLayout(3, 3));  for (char i = 'A'; i <= 'F'; i++) {  buttonPanel.add(new JButton("" + i));  }  buttonPanel.add(new JButton("0"));  buttonPanel.add(new JButton("Start"));  buttonPanel.add(new JButton("Stop"));  add(buttonPanel);  setTitle("Buttons");  setLocationRelativeTo(null);  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setSize(400, 250);  setVisible(true);  }  public static void main(String[] args) {  new ButtonsClass();  }  } |
| --- |

Sample Answer:



1. The role of layout managers is to arrange the UI components within the container

Sample Answer:



**Flow Layout (default to the JPanel)**

* The components are arranged in the container from left to right in the order they were added.

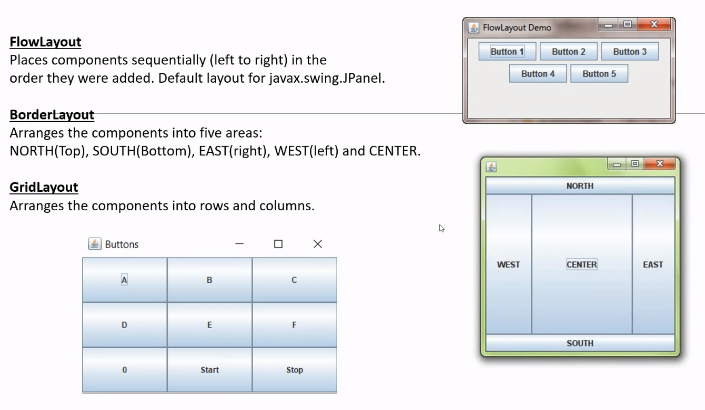
**Grid Layout**

* The components in a grid (matrix) formation with the number of rows and columns defined by the constructor

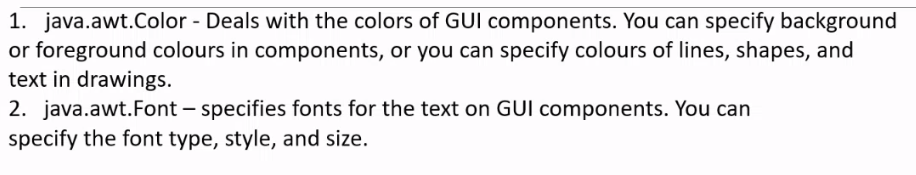
**Border Layout (default to the JFrame)**

* Divides the container into five areas: East, South, West, North, and Center and components are added to a BorderLayout by using the add(Component, index) method where index is one of the following constants.

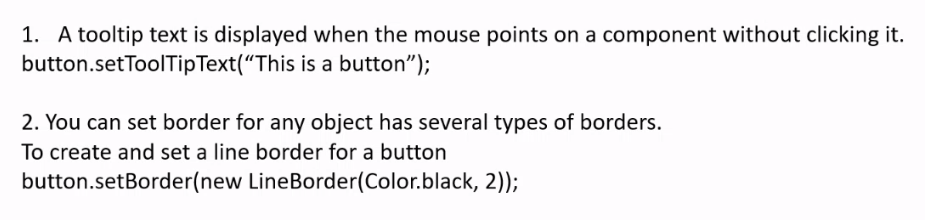
Sample Answer:



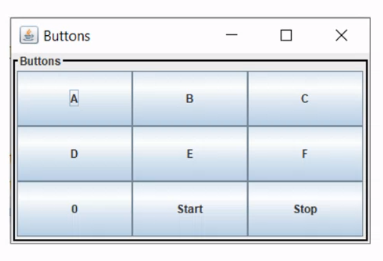
1. Sample Answer:



1. Sample Answer:



1. TitledBorder

Example:

* **Create new folder within that package**
* **Name it as “image”**
* **Put the image icon in that folder**

| import javax.swing.JFrame;  import javax.swing.ImageIcon;  public class TestImageIcon extends JFrame{    ImageIcon icon;    TestImageIcon(){  setTitle("TestImageIcon");  setLocationRelativeTo(null);  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setSize(400, 100);  setVisible(true);  icon = new ImageIcon("image/icon.png");  setIconImage(icon.getImage());  }    public static void main(String[] args) {  new TestImageIcon();  }  } |
| --- |

* **Follow method in Lecture Notes**

| import javax.swing.JFrame;  import javax.swing.ImageIcon;  public class TestImageIcon extends JFrame{    ImageIcon icon;    TestImageIcon(){  setTitle("TestImageIcon");  setLocationRelativeTo(null);  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setSize(400, 100);  setVisible(true);    icon = new ImageIcon(getClass().getResource("images/icon.png"));  add(new JLabel(icon));  }    public static void main(String[] args) {  new TestImageIcon();  }  } |
| --- |

Sample Answer:

