**Card Layout**

The CardLayout sets the components in a container as a stack of cards.It shows only one card (the card at the top) like a stack. It makes only one component visible at a time. It treats as if each component is a card that is why it is known as Card Layout.

**Two types constructors of CardLayout class**

**CardLayout():** creates a card layout with zero horizontal and vertical gap.

**CardLayout(int hgap, int vgap):** creates a card layout with the given horizontal and vertical gap.

**Some methods of CardLayout class**

**public void next(Container parent):** is used to flip to the next card of the given container.

**public void previous(Container parent):** is used to flip to the previous card of the given container.

**public void first(Container parent):** is used to flip to the first card of the given container.

**public void last(Container parent):** is used to flip to the last card of the given container.

**public void show(Container parent, String name):** is used to flip to the specified card with the given name.

**This is a manual code describe on Card Layout…**

package layout;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import java.awt.CardLayout;

public class CardLayoutExample implements ActionListener

{

JButton button[] = new JButton[5];

CardLayout cardLayout;

Container containerObject;

public CardLayoutExample()

{

makeCardLayout();

}

public void makeCardLayout()

{

JFrame frame = new JFrame("Card Layout Frame");

containerObject = frame.getContentPane();

cardLayout = new CardLayout(10,10);

containerObject.setLayout(cardLayout);

for(int i = 0; i<5; i++)

{

button[i] = new JButton("Student "+(i+1));

}

for(int i = 0; i<5; i++)

{

containerObject.add(button[i]);

}

for(int i = 0; i<5; i++)

{

button[i].addActionListener(this);

}

frame.setSize(400,400);

frame.setVisible(true);

}

@Override

public void actionPerformed(ActionEvent arg0)

{

cardLayout.next(containerObject); // For seeing the next card

}

public static void main(String[] args)

{

new CardLayoutExample();

}

} //end of the class