```java
import java.awt.*;
import javax.swing.*;
import javax.swing.border.*;

public class TopFiveDestinationList {
    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                TopDestinationListFrame topDestinationListFrame = new
TopDestinationListFrame();
                topDestinationListFrame.setTitle("Top 5 Destination
List");
                topDestinationListFrame.setVisible(true);
            }
        });
    }
}


class TopDestinationListFrame extends JFrame {
    private DefaultListModel listModel;

    public TopDestinationListFrame() {
        super("Top Five Destination List");

        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        setSize(900, 750);

        listModel = new DefaultListModel();


        //Make updates to your top 5 list below. Import the new image
files to resources directory.
        addDestinationNameAndPicture("1. Hawaii (It is a group of
beautiful islands in the Pacific Ocean.)", new
ImageIcon(getClass().getResource("/resources/https://
images.app.goo.gl/MoTMF7LPrZhj5kgG6")));
        addDestinationNameAndPicture("2. Bali (It is a paradise island
vacation destination in Indonesia.)", new
ImageIcon(getClass().getResource("/resources/https://
images.app.goo.gl/uNAjBda7EahRSVVP9")));
        addDestinationNameAndPicture("3. Dubai (It is famous for being
a popular luxury travel destination.)", new
ImageIcon(getClass().getResource("/resources/https://
images.app.goo.gl/DEG3r99rwcMcR5yr5")));
        addDestinationNameAndPicture("4. Italy (It is known for their
culture, art, film, and fashion.)", new
ImageIcon(getClass().getResource("/resources/https://
images.app.goo.gl/jq8Te1zkhKFn9HQ5A")));
        addDestinationNameAndPicture("5. France (It is a home to
vibrant cities, landscapes, food and culture.)", new
ImageIcon(getClass().getResource("/resources/https://
images.app.goo.gl/ifAe3xvhFvWX5zxW8")));

        JList list = new JList(listModel);
        JScrollPane scrollPane = new JScrollPane(list);

        TextAndIconListCellRenderer renderer = new
TextAndIconListCellRenderer(2);

        list.setCellRenderer(renderer);

        getContentPane().add(scrollPane, BorderLayout.CENTER);
    }

    private void addDestinationNameAndPicture(String text, Icon icon)
{
        TextAndIcon tai = new TextAndIcon(text, icon);
        listModel.addElement(tai);
    }
}


class TextAndIcon {
    private String text;
    private Icon icon;

    public TextAndIcon(String text, Icon icon) {
        this.text = text;
        this.icon = icon;
    }

    public String getText() {
        return text;
    }

    public Icon getIcon() {
        return icon;
    }

    public void setText(String text) {
        this.text = text;
    }

    public void setIcon(Icon icon) {
        this.icon = icon;
    }
}
```

```
class TextAndIconListCellRenderer extends JLabel implements
ListCellRenderer {
    private static final Border NO_FOCUS_BORDER = new EmptyBorder(1,
1, 1, 1);

    private Border insideBorder;

    public TextAndIconListCellRenderer() {
        this(0, 0, 0, 0);
    }

    public TextAndIconListCellRenderer(int padding) {
        this(padding, padding, padding, padding);
    }

    public TextAndIconListCellRenderer(int topPadding, int
rightPadding, int bottomPadding, int leftPadding) {
        insideBorder = BorderFactory.createEmptyBorder(topPadding,
leftPadding, bottomPadding, rightPadding);
        setOpaque(true);
    }

    public Component getListCellRendererComponent(JList list, Object
value,
    int index, boolean isSelected, boolean hasFocus) {
        // The object from the combo box model MUST be a TextAndIcon.
        TextAndIcon tai = (TextAndIcon) value;

        // Sets text and icon on 'this' JLabel.
        setText(tai.getText());
        setIcon(tai.getIcon());

        if (isSelected) {
            setBackground(list.getSelectionBackground());
            setForeground(list.getSelectionForeground());
        } else {
            setBackground(list.getBackground());
            setForeground(list.getForeground());
        }

        Border outsideBorder;

        if (hasFocus) {
            outsideBorder =
UIManager.getBorder("List.focusCellHighlightBorder");
        } else {
            outsideBorder = NO_FOCUS_BORDER;
        }

        setBorder(BorderFactory.createCompoundBorder(outsideBorder,
insideBorder));
        setComponentOrientation(list.getComponentOrientation());
        setEnabled(list.isEnabled());
        setFont(list.getFont());

        return this;
    }

    // The following methods are overridden to be empty for
performance
    // reasons. If you want to understand better why, please read:
    //
    // http://java.sun.com/javase/6/docs/api/javax/swing/
DefaultListCellRenderer.html#override

    public void validate() {}
    public void invalidate() {}
    public void repaint() {}
    public void revalidate() {}
    public void repaint(long tm, int x, int y, int width, int height)
{}
    public void repaint(Rectangle r) {}
}
```