Unit 12: Introduction to Java Applets [1 Hr.]

1. Definition of Java Applets

Definition: A Java applet is a small application written in Java that can be embedded into a web page. Applets run within a web browser using the Java Plugin. They are used to create dynamic and interactive content on web pages.

Key Characteristics:

- Applets are executed on the client side (in the browser).
- They are embedded in HTML using the `<applet>` or `<object>` tag.
- Applets are platform-independent and rely on the Java Runtime Environment (JRE).

---

2. Applet Lifecycle Methods

An applet has a well-defined lifecycle, controlled by the following methods:

1. **`init()`**:
    - Called when the applet is first loaded.
    - Used for initialization tasks like setting up variables or loading resources.
2. **`start()`**:
    - Called after `init()` and whenever the applet is restarted (e.g., when the user returns to the web page).
    - Used to start or resume the applet's execution.
3. **`paint(Graphics g)`**:
    - Called whenever the applet needs to redraw its output (e.g., when the window is resized).
    - Used to draw graphics or text on the applet's surface.
4. **`stop()`**:
    - Called when the applet is stopped (e.g., when the user navigates away from the web page).
    - Used to pause or stop the applet's execution.
5. **`destroy()`**:
    - Called when the applet is about to be unloaded (e.g., when the browser is closed).
    - Used to clean up resources.

---

3. Building a Simple Applet

Lab 1: Building a Simple Applet:

```
import java.applet.Applet;
import java.awt.Graphics;

public class SimpleApplet extends Applet {
```

```
    // Lifecycle method: Called when the applet is initialized
    public void init() {
        setBackground(Color.cyan); // Set background color
    }

    // Lifecycle method: Called to draw the applet's content
    public void paint(Graphics g) {
        g.drawString("Hello, Applet!", 50, 50); // Draw text on the applet
    }
}
```

Explanation:

- We create a class `SimpleApplet` that extends the `Applet` class.
- The `init()` method sets the background color of the applet.
- The `paint()` method is overridden to display the text "Hello, Applet!" at coordinates (50, 50).

---

4. Using Applet Viewer

Definition: The **Applet Viewer** is a tool provided by the JDK to test and run applets without a web browser. It simulates the environment of a web browser for applet execution.

Steps to Use Applet Viewer:

1. Save the applet code in a file (e.g., `SimpleApplet.java`).
2. Compile the applet using the command:

   ```
   javac SimpleApplet.java
   ```

3. Create an HTML file to embed the applet (e.g., `SimpleApplet.html`): **Lab 1 continue**

   ```
   <html>
     <body>
    <applet code="SimpleApplet.class" width="300" height="200"></applet>
     </body>
   </html>
   ```

4. Run the applet using the Applet Viewer:

   ```
   appletviewer SimpleApplet.html
   ```

Sample Output: The Applet Viewer opens a window displaying the applet with the text "Hello, Applet!" on a cyan background.

---

5. Adding Controls to Applets

Definition: We can add GUI components like buttons, text fields, and labels to applets to make them interactive.

Lab 2: Adding Controls to Applets:

```java
import java.applet.Applet;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class ControlApplet extends Applet implements ActionListener {
    Button button;
    TextField textField;

    // Lifecycle method: Called when the applet is initialized
    public void init() {
        button = new Button("Click Me"); // Create a button
        textField = new TextField(20); // Create a text field
        add(button); // Add button to the applet
        add(textField); // Add text field to the applet
        button.addActionListener(this); // Register button for event handling
    }

    // Event handler for button click
    public void actionPerformed(ActionEvent e) {
        textField.setText("Button Clicked!");
    }
}
```

Explanation:

- We add a `Button` and a `TextField` to the applet.
- The `actionPerformed()` method is called when the button is clicked, and it updates the text field with the message "Button Clicked!".

Sample Output: The applet displays a button and a text field. When the button is clicked, the text field displays "Button Clicked!".

---

6. Animation Concepts in Applets

Definition: Animation in applets involves displaying a sequence of images or shapes in rapid succession to create the illusion of movement.

Lab 3: Animation Concepts in Applets:

```java
import java.applet.Applet;
import java.awt.*;

public class AnimationApplet extends Applet implements Runnable {
    int x = 0; // X-coordinate of the circle
    Thread animationThread;

    // Lifecycle method: Called when the applet is initialized
    public void init() {
        setBackground(Color.white);
    }

    // Lifecycle method: Called when the applet is started
```

```
    public void start() {
        animationThread = new Thread(this); // Create a new thread for animation
        animationThread.start(); // Start the thread
    }

    // Thread's run method
    public void run() {
        while (true) {
            x += 5; // Move the circle to the right
            if (x > getWidth()) {
                x = 0; // Reset the circle's position
            }
            repaint(); // Redraw the applet
            try {
                Thread.sleep(100); // Pause for 100 milliseconds
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }

    // Lifecycle method: Called to draw the applet's content
    public void paint(Graphics g) {
        g.setColor(Color.red);
        g.fillOval(x, 50, 50, 50); // Draw a circle at (x, 50)
    }
}
```

Explanation:

- We create a simple animation where a red circle moves horizontally across the applet.
- The `run()` method updates the circle's position and calls `repaint()` to redraw the applet.
- The `Thread.sleep(100)` method creates a delay of 100 milliseconds between frames.

Sample Output: The applet displays a red circle that moves from left to right across the screen.

---

7. Differences Between Applets and Applications

Feature	Applets	Applications
Execution	Runs within a web browser.	Runs as a standalone program.
Security	Subject to strict security restrictions (sandbox).	No security restrictions.
Lifecycle	Has a well-defined lifecycle (init, start, stop, destroy).	No predefined lifecycle.
Usage	Used for web-based interactive content.	Used for general-purpose programs.

Summary

In this unit, we explored **Java Applets**, which are small programs designed to run within web browsers. We learned about the applet lifecycle methods (`init()`, `start()`, `paint()`, `stop()`, and `destroy()`), built a simple applet, and used the **Applet Viewer** for testing. We also added controls like buttons and text fields to make applets interactive and implemented basic animation concepts.

By understanding applets, we can create dynamic and interactive content for web pages. However, it's important to note that modern web technologies like HTML5 and JavaScript have largely replaced applets for web-based interactivity. Nonetheless, applets remain a valuable part of Java's history and provide a foundation for understanding GUI programming in Java.