# JAYOTI VIDYAPEETH WOMEN'S UNIVERSITY, JAIPUR

## (Format for Preparing E Notes)

## Faculty of CSE

**Faculty Name-** JV'n Ayushi Choudhary (Assistant Professor/ Associate Professor/ Professor)

**Program- ...........**BCA 5$^{th}$ Semester

**Course Name** – Advanced Java

**Academic Day starts with –**

- Greeting with saying **'Namaste'** by joining Hands together following by 2-3 Minutes Happy session, Celebrating birthday of any student of respective class and **National Anthem**.

- Lecture Starts with-

- Review of previous Session- Working with AWT

- Topic to be discussed today : Working with Windows

Working with windows in Java typically involves creating and managing graphical user interface (GUI) windows for desktop applications. Java provides different libraries and frameworks for GUI development, including AWT (Abstract Window Toolkit), Swing, and Java FX. Short note on working with windows in Java:

**1. Choosing a GUI Library:** In Java, there are several options for creating windows. AWT is the basic GUI library, Swing is an extension of AWT with more components and features, and Java FX is a modern, versatile GUI framework introduced in Java 8.

**2. Creating a Window:** In AWT, window is created using the **Frame** or **Window** classes. In Swing, **J Frame** is commonly used, and in Java FX, it works with a **Stage** and **Scene**. Instantiate the window object, set its properties (e.g., title, size), and configure its layout.

**3. Adding Components:** To make the window interactive, add components like buttons, labels, text fields, and other widgets to it. Different libraries provide a wide variety of components for building user interfaces.

**4. Event Handling:** Windows and their components generate events when users interact with them. Event listeners are used to capture and respond to these events. In Java, event listeners are created by implementing interfaces or using adapters, depending on the library working with.

**5. Layout Management:** Effective layout management is essential for arranging and sizing components within the window. AWT, Swing, and Java FX offer layout managers for this purpose. Layout managers automatically handle the placement of components based on the window's size.

**6. Graphics and Custom Drawing:** Many GUI applications require custom graphics. Java provides methods for custom drawing within a window, allowing you to create shapes, charts, and complex visual elements.

**7. Window Decorations:** Windows typically have standard decorations, such as title bars, close buttons, and borders. These can be customized to some extent depending on the library and platform.

**8. Event Loop and Threading:** GUI applications run an event loop to handle user interactions and events. It's essential to manage this loop properly, and Java applications often use multiple threads to separate the UI thread from background tasks to maintain responsiveness.

**9. Platform Independence:** Java's GUI libraries are designed to be platform-independent, allowing application to run on different operating systems without modification.

**10. Window Management:** Operating systems provide features like window minimizing, maximizing, and taskbar management, which can be accessed through the GUI library's API.

In summary, working with windows in Java involves selecting the appropriate GUI library (AWT, Swing, or Java FX), creating windows, adding components, handling events, and ensuring effective layout management. Properly designed and managed windows provide the foundation for building interactive and user-friendly desktop applications in Java.