# Unit - 3 I/O Streams / Applet

Java streams provide many **benefits** that make I/O operations easier, more efficient, and flexible:

1. **Abstraction**: Stream classes abstract data sources and destinations, allowing easier interaction with various data formats.
2. **Efficiency**: Buffered streams and other features improve performance and reduce the number of I/O operations.
3. **Flexibility**: Support for multiple types of data (e.g., byte, character, object) and a wide range of I/O operations.
4. **Convenience**: Higher-level classes (e.g., PrintWriter, Scanner) offer simple methods for handling complex tasks like formatted output.
5. **Error Handling**: Automatic resource management (try-with-resources) reduces the risk of resource leaks and simplifies error handling.
6. **Object Serialization**: Easily store and retrieve Java objects.
7. **Functional Style**: The Stream API provides a declarative way to process sequences of data with operations like map, filter, and reduce.

<applet code="HelloWorld" width=200 height=60>
</applet>

| Parameters | Java Application | Java Applet |
|---|---|---|
| Definition | Applications are just like a Java program that can be executed independently without using the web browser. | Applets are small Java programs that are designed to be included with the HTML web document. They require a Java-enabled web browser for execution. |

| | | |
|---|---|---|
| main () method | The application program requires a main() method for its execution. | The applet does not require the main() method for its execution instead init() method is required. |
| Compilation | The “javac” command is used to compile application programs, which are then executed using the “java” command. | Applet programs are compiled with the “javac” command and run using either the “appletviewer” command or the web browser. |
| File access | Java application programs have full access to the local file system and network. | Applets don’t have local disk and network access. |
| Access level | Applications can access all kinds of resources available on the system. | Applets can only access browser-specific services. They don’t have access to the local system. |
| Installation | First and foremost, the installation of a Java application on the local computer is required. | The Java applet does not need to be installed beforehand. |
| Execution | Applications can execute the programs from the local system. | Applets cannot execute programs from the local machine. |

| | | |
|---|---|---|
| Program | An application program is needed to perform some tasks directly for the user. | An applet program is needed to perform small tasks or part of them. |
| Run | It cannot run on its own; it needs JRE to execute. | It cannot start on its own, but it can be executed using a Java-enabled web browser. |
| Connection with servers | Connectivity with other servers is possible. | It is unable to connect to other servers. |
| Read and Write Operation | It supports the reading and writing of files on the local computer. | It does not support the reading and writing of files on the local computer. |
| Security | Application can access the system's data and resources without any security limitations. | Executed in a more restricted environment with tighter security. They can only use services that are exclusive to their browser. |
| Restrictions | Java applications are self-contained and require no additional security because they are trusted. | Applet programs cannot run on their own, necessitating the maximum level of security. |

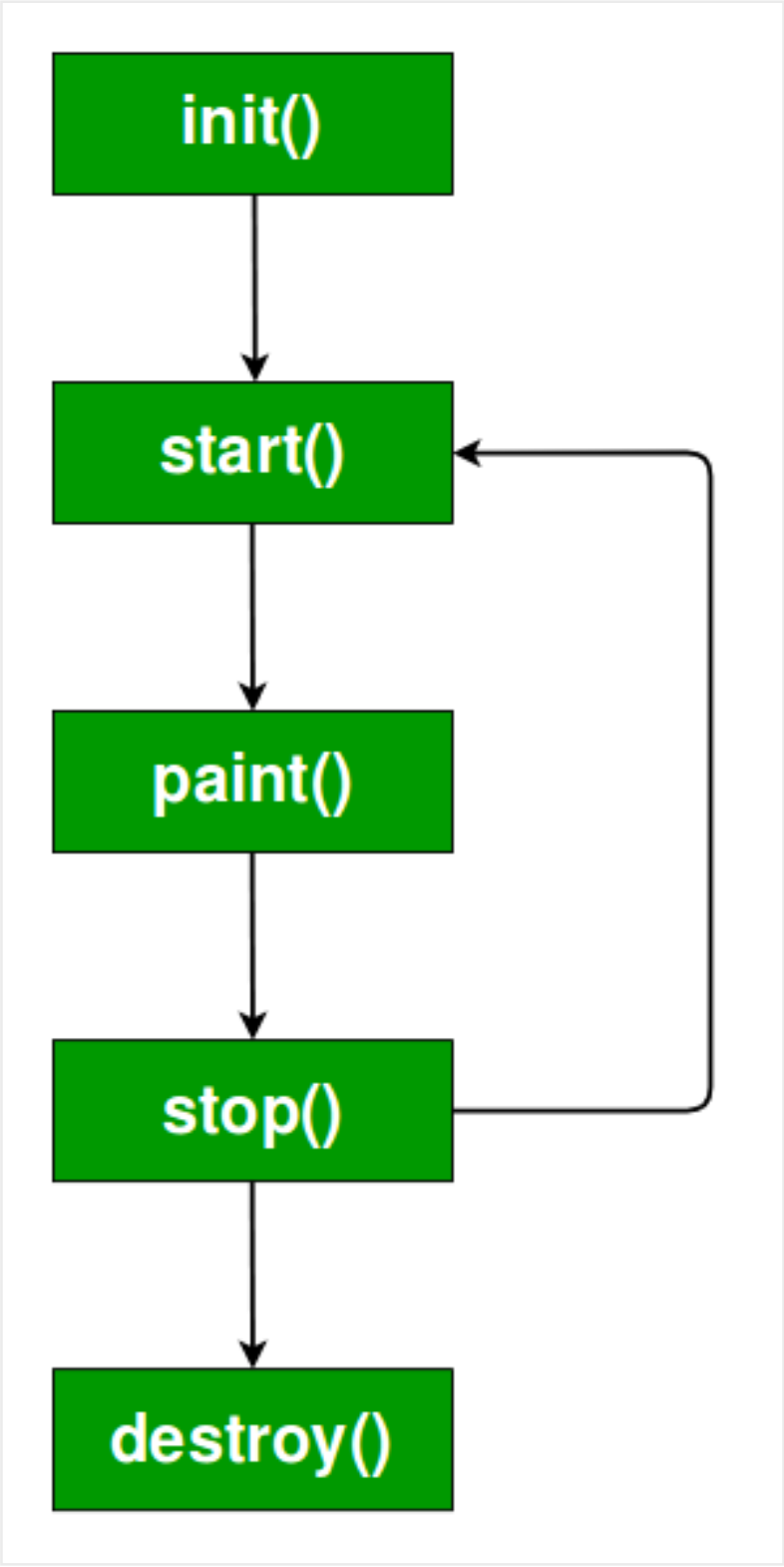
## Creating Applets in Java

An **applet** is a small application that runs within a web browser or applet viewer. It is written in Java and is used to provide interactive content, animations, or games embedded into a web page. Applets have largely been deprecated and are no longer widely used, as modern web technologies like JavaScript and HTML5 have taken over

interactive tasks. However, understanding how applets work can still be useful for historical context or legacy applications.

**Applet Basics**

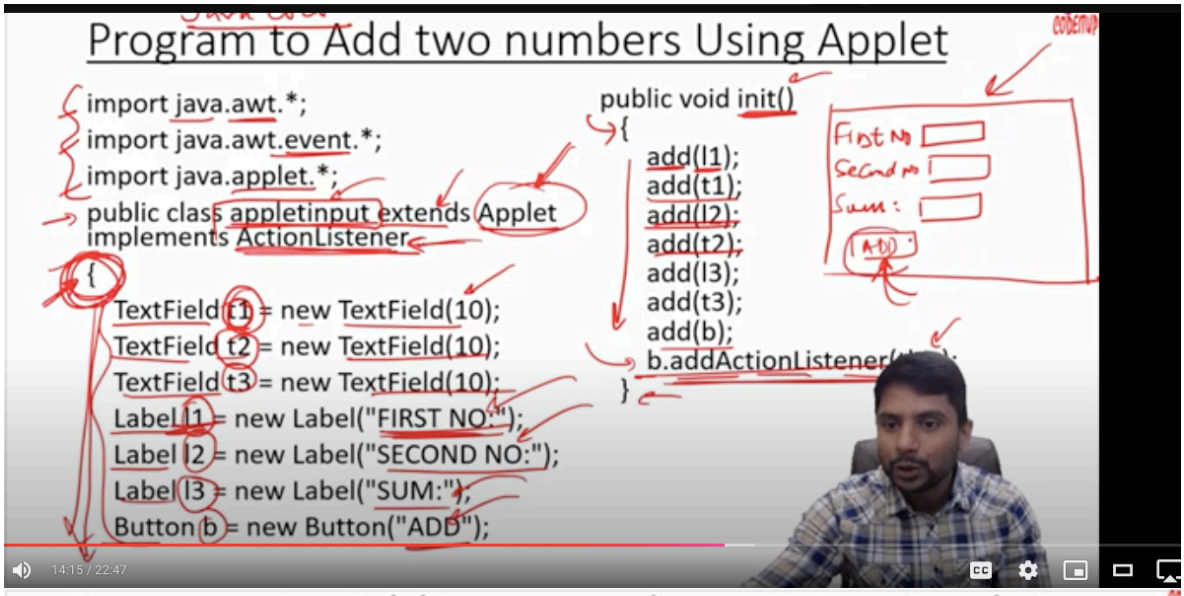
- An **applet** is a Java class that extends the java.applet.Applet class (or javax.swing.JApplet for Swing-based applets).
- Applets are designed to run inside a web browser or in an applet viewer.
- Applets do not have a main() method. Instead, they rely on lifecycle methods that the browser or applet viewer calls during the applet's execution.

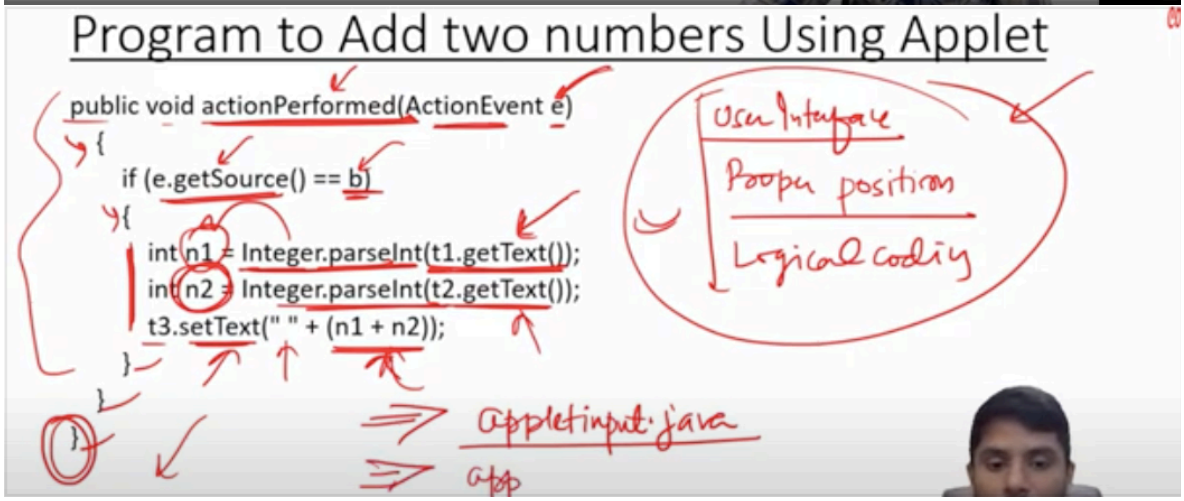


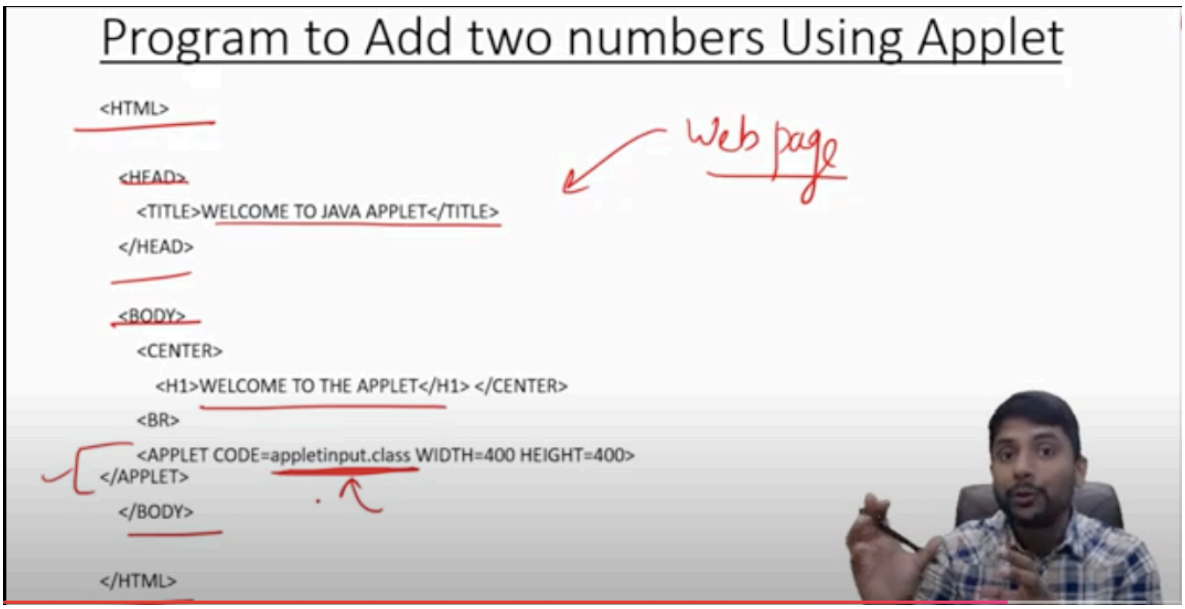

**Summary**

1. **Applet Basics**: An applet is a small Java program that runs inside a browser or applet viewer. It has no main()method and relies on lifecycle methods like init(), start(), stop(), and destroy().
2. **Applet Architecture**: An applet must extend java.applet.Applet and is embedded into an HTML page using

the <applet> tag.

3. **Life Cycle**: The applet goes through various phases (initialization, execution, stop, and destruction), with methods corresponding to each phase.
4. **Display Methods**: paint(Graphics g) is used for drawing content. The applet can request a repaint with repaint().
5. **Requesting Repainting**: repaint() triggers a re-call of the paint() method, and you can override update() for custom painting logic.
6. **Status Window**: Use showStatus() to display messages in the applet's status window.
7. **HTML <APPLET> Tag**: Used to embed the applet in an HTML page and pass parameters to the app
8. 

9. 




## AWT Classes:

AWT provides classes like Frame, Window, and Dialog for creating windows.

**Creating Windows**: A Frame is a common window container in AWT, created with new Frame().

**Displaying Windows**: Set the size using setSize() and visibility using setVisible().

**Handling Events**: Use event listeners (e.g., ActionListener, WindowListener) to handle user interactions.

**Frame in Applet**: A Frame can be created within an applet, but applets themselves usually run within a browser or applet viewer.