# ADVANCED JAVA LAB (23IT5351)

**Date:** 23-6-2025 **Roll No:**238W1A1296

## Week – 1: Foundation to Advanced Java Lab - Preparation Lab.

### 1. What are the differences between Core Java and Advanced Java?

Core Java focuses on the fundamental building blocks and syntax of the Java language, while Advanced Java builds upon these basics to create complex, enterprise-level applications.

**Core java:**

- Covers the core syntax, data types, object-oriented principles (classes, inheritance, polymorphism, encapsulation), exception handling, and basic input/output operations.
- Primarily used for developing general-purpose, standalone applications, like desktop applications, simple games, or utilities.
- **Ex:** Basic applications, event management systems, graphic video games, calculators.

**Advanced Java:**

- Focuses on building complex, enterprise-level applications, including web applications, distributed systems, and enterprise-level solutions.
- Utilizes technologies like JDBC for database connectivity, Servlets and JSPs for web development, web services, and frameworks like Spring and Hibernate.
- Used for developing web applications, enterprise-level applications, and applications requiring complex functionality and scalability.
- **Ex:** Web applications, e-commerce platforms, banking applications, and other large-scale enterprise systems.

### 2. What are the main important features of Applets in Java?

- Applets run on any system with a Java Virtual Machine (JVM), enabling cross-platform compatibility.
- Applets are embedded in HTML and run within web browsers, providing interactive content.
- Applets operate in a sandboxed environment, restricting access to system resources for enhanced security.
- They allow for dynamic user interfaces, animations, games, and real-time collaboration features.
- Methods such as init(), start(), stop(), and destroy() manage the applet's lifecycle.

### 3. What are the key features of Java Servlets?

- Servlets run on the server, processing client requests and generating dynamic web content.
- Written in Java, servlets are portable across platforms.
- Servlets are efficient and scalable, handling multiple requests via multithreading.

- They integrate seamlessly with databases and other backend systems.
- Methods like init(), service(), and destroy() manage servlet lifecycles.
- Servlets can track user sessions for stateful web applications.

**4. What are the differences between Applets and Servlets in Java?**

- **Applets:**
  - Executed on Client side (in browser).
  - Purpose of usage is to enhance page interactivity.
  - Applets use user interface classes like AWT and Swing.
  - Applets are more prone to risk as it is on the client machine.
  - Requires java compatible browser for execution.
- **Servlets:**
  - Executed on server side (on web server).
  - Purpose of usage is to process requests, generate dynamic pages.
  - No User interface required.
  - Servlets are under the server security.
  - It accepts input from browser and generates response in the form of HTML Page, JavaScript Object, Applets etc.

**5. What are the major features of Java compared to other programming languages?**

- Java's "write once, run anywhere" philosophy through the JVM.
- OOP principles are strongly supported.
- Exception handling, a security manager, and automatic memory management (garbage collection).
- Concurrent programming is supported by default, and there are numerous APIs for networking, data structures, graphical user interfaces, database connectivity, and other areas.
- Reflection and dynamic class loading are supported.

**6. How do JDBC and ODBC primarily contribute to database connectivity?**

- **Java Database Connectivity**, or JDBC, offers a common API through which Java apps can communicate with databases, facilitating database functions such as data querying and updating.
- **ODBC (Open Database Connectivity):** Enables applications to connect to a variety of databases by serving as a standard API for database access across platforms and programming languages.

**7. What are the differences between ODBC and JDBC?**

- **ODBC:**
  - It is language independent.
  - C, C++, Python, Perl, and other programming languages can all be used with this language-independent API.

- It serves as a link between the database and the application. It needs a database driver to communicate. Usually, database vendors supply ODBC drivers.

- **JDBC:**
  - It is Java-specific, works in Java applications.
  - It supports only Java language.
  - It connects Java applications to the database using JDBC drivers. JDBC is built to be simpler for Java developers and offers direct database access within the Java environment.

**8. What are the differences between Cookies and Sessions in web applications?**

- **Cookies:** Small pieces of data stored on the client's browser. They help remember user information between requests. They are less secure and users can disable them.
- **Sessions:** Server-side storage of user data. A session ID is stored in a cookie or URL. Sessions are more secure and can hold larger amounts of data.

**9. What are the differences between Spring Boot and the traditional Spring Framework?**

- **Spring Framework** requires manual configuration for most components, such as beans and data sources, and is very flexible.
- **Spring Boot** offers auto-configuration, embedded servers, and strong defaults to make setup and development easier. It cuts down on boilerplate code and is made for quick application development.

**10. What are the differences between standalone applications and web applications?**

**Standalone Applications:**

- Installed on a local machine.
- Specific to a platform (for example, Windows or macOS).
- Needs to be installed.
- Updates are done manually.
- Can be used offline and faster since it runs locally.
- Direct access to system resources.
- Relies on local security as maintained by the user.
- Only available on the installed device since it is stored locally.
- **Examples:** MS Word, Photoshop.

**Web Applications:**

- Accessed through a browser and hosted on a server.
- Works on any platform so no need of installation.
- Updates happen automatically so it needs internet access.
- Depends on the network or server.
- Limited access to local resources.
- Uses web-based security measures as maintained by the server.

- Can be accessed from anywhere as stored on a remote server or in the cloud.
- **Examples:** Gmail, Facebook.

**11. What are the basic requirements for executing a web application?**

- **Web Server:** For hosting and serving the application (e.g. Tomcat, Nginx, Apache).
- **Application Server (may be needed):** for dynamic content and business logic (e.g. Tomcat, WildFly).
- **Database (may be needed):** for storing data and retrieving data.
- **Client browser:** To access and interact with the web application.
- **Network connection:** For client-server communication.
- **Backend technology:** (server-side processing) Java Servlets, JSP, PHP, or Node.js.
- **Frontend technology:** HTML, CSS, JavaScript.