

# Chapter 1: Introduction to Applications

---

## Q1. - Define Applications.

**Answer:** An application is a collection of related programs that are designed to perform specific tasks or operations for the user. These programs work together in a coordinated manner to achieve a particular functionality, such as managing data, processing information, or interacting with users.

### Applications can be:

Stand Alone/Desktop Applications – Run on personal computers (e.g., Microsoft Word, Notepad).

Web Applications – Run in web browsers and are accessed over the internet (e.g., Gmail, Facebook).

Mobile Applications – Designed for mobile devices (e.g., WhatsApp, Instagram).

Enterprise Applications – Large-scale software used in organizations (e.g., ERP, CRM systems).

## Q. How do we give input?

**Answer:** In application development, inputs can be provided through different levels or mediums depending on the type of application and its interface. The main four input levels are:

- using Console Input
- using GUI Input
- WebBroswer Input
- Application to Application(Websservices)

### 1. Console Input:

-----

**Definition:** The most basic form of input where users provide data through the terminal or command-line interface.

**Example:** Java's Scanner or BufferedReader classes to read input from the keyboard.

**Use Case:** Used in beginner-level Java programs, utility scripts, or system-level tools.

e.g:

```
Scanner sc = new Scanner(System.in);  
System.out.print("Enter your name: ");  
String name = sc.nextLine();
```

### 2. GUI Input (Graphical User Interface):

-----

**Definition:** Input is taken via graphical components like text fields, buttons, checkboxes, etc.

**Tools/Technologies:** Java Swing, JavaFX, AWT.

Use Case: Desktop applications with a user-friendly interface.

e.g:

```
TextField nameField = new TextField();  
JButton submit = new JButton("Submit");
```

### 3. Web Browser Input:

Definition: Input is provided through HTML forms and web elements within a web browser.

Tools/Technologies: HTML, CSS, JavaScript on the front end; Servlets, JSP, Spring MVC on the back end.

Use Case: Web-based applications accessed via browsers.

e.g:

```
<form action="submitData" method="post">  
  <input type="text" name="username" />  
  <input type="submit" value="Submit" />  
</form>
```

### 4. Application to Application (Web Services):

Definition: Input and data exchange happen between applications without human involvement, often over a network.

Technologies: REST APIs, SOAP, JSON, XML, HTTP/HTTPS.

Use Case: Microservices, cloud communication, payment gateways, third-party integrations.

e.g:

```
// JSON input sent from one app to another  
{  
  "userId": "12345",  
  "action": "login"  
}
```

### Q.3. What are the types of applications? Explain.

**Answer:** Applications are categorized based on their platform and usage.

Below are the common types of applications:

#### 1. Desktop Applications/ Stand Alone Applications:

Definition: These are software programs that run locally on a personal computer or laptop. They don't require a web browser or an internet connection(except for updates or online features).

- These applications are also termed as non server application.
- SAA are categorised into two types based on user's interaction:
  - CUI Application: user interacts through console.
  - GUI Application: user interacts through GUI components.  
(Swing,JavaFx,AWT used for GUI applications)

Features:

- Installed directly on a system.
- Can work offline.
- Utilizes the computer's hardware resources.

Examples:

Microsoft Word – A word processing application.

Notepad – A basic text editor for writing and editing plain text.

## 2. Web Applications :

---

Definition: These applications run in a web browser and are accessed through the internet using URLs.

These application which are executed in the server environment and interacts with the user through web browser.

They collect data from browsers.

- These applications can be developed using components from Adv. Java:

- **JDBC:** Java Database Connectivity

- used to interact with database product.

- **Servlet:** Server Program which accepts the requests from browser and then provides the response.

- **JSP:** Java Server Page:

It gives response from web application.

Features:

- No installation required on the client-side.
- Centralized updates and maintenance.
- Accessible from any device with internet and a browser.

Examples:

Gmail – For sending and receiving emails.

Facebook – A social networking platform.

## 3. Mobile Applications:

---

Definition: Apps developed specifically for mobile devices like smartphones and tablets.

They can be native (platform-specific), hybrid (combination of

Features:

- Installed through app stores (e.g., Google Play Store, Apple App Store).
- Can access device features like camera, GPS, etc.
- May work offline or require internet.

Examples:

WhatsApp – A messaging and calling app.

Instagram – A photo and video sharing platform.

## 4. Enterprise Applications :

---

Definition: These are large-scale software systems designed to operate in a corporate environment. They help manage business processes and data across departments.

Features:

- Complex, scalable, and secure.
- Supports multiple users and roles.
- Often integrates with other systems.

Examples:

ERP (Enterprise Resource Planning)– Helps in managing business processes

like inventory, finance, HR (e.g., SAP).

CRM (Customer Relationship Management) – Manages customer data and interactions (e.g., Salesforce).

Adv. Java by Jagannath