

Experiment No. 9

Designing the GUI using swing components & IDE.

Instructions:

This manual consists of three parts:

- A) Theory and Concepts,
- B) Problems for Implementation, and
- C) Write-up Questions.

1. Students must understand the **theory and concepts** provided before implementing the problem statement(s) for **Experiment 9**.
2. They should **practice the given code snippets** within the theory section.
3. Later, they need to **implement the problems provided**.
4. **Write-up:** Students are required to **write answers** to the questions on journal pages, **maintain a file**, and get it checked regularly. The file should include index, write-up, and implementation code with results.
5. **Referencing:** Include proper sources or references for the content used.
6. Use of Generative AI: Clearly mention if you have used any AI tools (e.g., ChatGPT, Copilot, Gemini) to generate text, explanations, or code. Cite the AI-generated content appropriately in the write-up.

Part A. Theory and Concepts:

There are various IDEs available for Java development, each offering unique features to enhance coding efficiency. Below are four popular choices:

1. IntelliJ IDEA (By JetBrains):

IntelliJ IDEA, developed by JetBrains, is a powerful and intelligent IDE known for its deep code understanding, developer-friendly UI, and productivity-enhancing features. It is available in two versions: Community (Free) and Ultimate (Paid).

- Smart Code Assistance: Intelligent auto-completion, refactoring, and inspections.
- Built-in Tools: Supports Git, database management, build tools (Maven, Gradle), and testing frameworks.
- Performance Optimized: Fast indexing, smooth navigation, and modern UI.
- Supports: Spring, JavaFX, Hibernate, Android, and more.

- **Install IntelliJ IDEA** - <https://www.youtube.com/watch?v=oJ9OdOgdTIg>
- **Official Documentation** - <https://www.jetbrains.com/help/idea/getting-started.html>

2. NetBeans (By Apache):

NetBeans is an open-source **Integrated Development Environment (IDE)** for Java that enables developers to build applications using a set of modular software components known as modules.

It is a cross-platform IDE that runs on **Windows, macOS, Linux, and Solaris**.

- **Simple & Cross-Platform:** Runs on Windows, macOS, Linux, Solaris.
- **Powerful Code Editor:** Supports Java SE, Java EE, JavaFX, PHP, HTML5, C/C++.
- **Rich Development Features:** Includes code analyzers, debugging tools, and NetBeans Profiler.
- **Install NetBeans** - <https://www.youtube.com/watch?v=45DcRMeLweE>
- **Official Documentation** - <https://netbeans.apache.org/tutorial/main/kb/docs/java-se/>

3. Eclipse (By Eclipse Foundation):

Eclipse is a widely used **open-source IDE** for Java development, known for its extensibility through plugins. It provides a flexible workspace and supports multiple programming languages, making it a powerful choice for developers.

- **Highly Extensible:** Supports Java, C/C++, Python, PHP, JavaScript, and more via plugins.
- **Specialized Development Environments:** Includes JDT for Java, CDT for C++, PDT for PHP.
- **Large Community & Open Source**
- **Install Eclipse** - <https://www.youtube.com/watch?v=LaxLZiV4mpM>
- **Official Documentation** - <https://docs.eclipse.xyz/>

4. Visual Studio Code (VS Code) (By Microsoft):

- **Best for:** Lightweight Java development with powerful extensions.
- **Extensive Plugin Support:** Java, Python, JavaScript, TypeScript.
- **Fast & Lightweight:** Less resource-intensive than full IDEs.
- **Built-in Terminal & Git Support**
- **Java Development via Extensions:** Debugger, IntelliSense, and Maven/Gradle integration.

- Install VS Code - https://www.youtube.com/watch?v=cu_ykIfBprl
- Official Documentation - <https://code.visualstudio.com/docs>

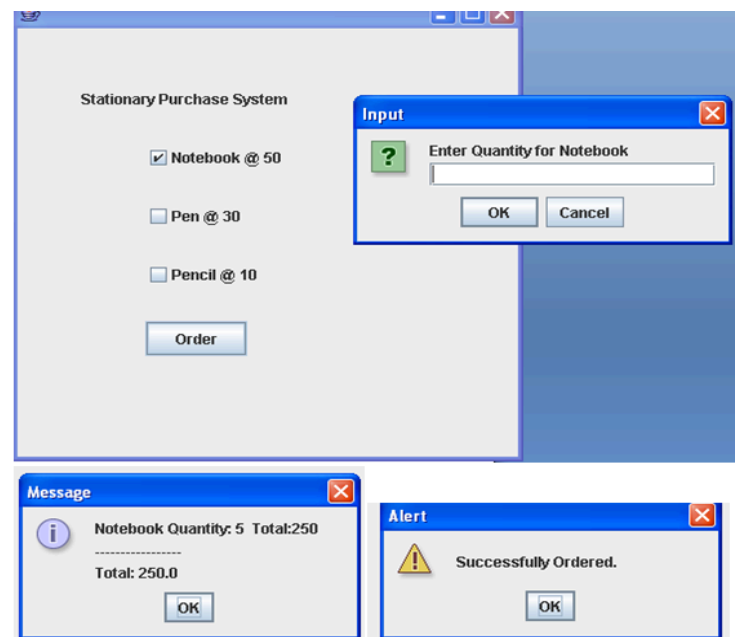
Part B. Problems for Implementation:

Aim: Designing the GUI using swing components & IDE.

1. Design a standard calculator using Swing components that supports basic operations (Addition, Subtraction, Multiplication, and Division). Implement this with IntelliJ IDEA

Implementation Guidelines:

- Use **JTextField** to display input/output.
 - Use **JButton** for digits (0-9) and operations (+, -, *, /, =, %, square, square-root, cube, C, etc.).
 - Implement event handling for button clicks.
 - Display results in the text field.
2. Implement the following problem statement using IntelliJ IDEA.



Part C. Write-up Questions:

1. Explain following Java modern tools / IDEs with their features - IntelliJ IDEA, NetBeans, Eclipse Visual Studio Code

Conclusion: After completing this exercise, students should be able to design the GUI for any application using the swing components.