

Introductory Session on Exploring Java Using NetBeans

LAB # 01

Lab Objective:

Objective of this lab is to make students understand the primary objectives are to acquaint students with the NetBeans Integrated Development Environment (IDE) for Java programming, teach them about variables in Java and how to use them effectively, introduce the fundamental concepts of decision-making through if statements, and provide a grasp of looping structures like for and while loops. Students will engage in practical exercises to apply these concepts, fostering problem-solving skills and enabling them to create basic Java programs involving variables, decision-making, and looping constructs.

Activity Outcomes:

Upon completing this lab, students will be able to proficiently utilize NetBeans for Java development, demonstrate competence in handling variables, implement decision-making constructs, and effectively utilize looping structures in Java programs.

Instructor Note:

The students should have knowledge about JAVA compiler, variables and functions.

Introduction:

Java is a versatile and widely used programming language renowned for its platform independence and robustness. In this lab, students will embark on their journey into the world of Java, gaining a foundational understanding of its syntax and principles. They will learn how Java's portability makes it suitable for diverse applications, from web development to mobile app creation and more. This introduction will set the stage for students to harness the power of Java in subsequent programming tasks.

In this lab, students will not only get acquainted with Java but also explore its practical implementation using the NetBeans Integrated Development Environment (IDE). NetBeans simplifies Java development by providing a user-friendly interface and a wealth of helpful features. Students will learn how to create, compile, and run Java programs within the NetBeans environment, setting the stage for productive coding sessions.

Furthermore, students will delve into essential programming concepts, including variables, decision-making constructs (such as if statements), and looping structures (such as for and while loops). They'll understand the significance of variables for storing and manipulating

data, grasp how decision-making constructs enable programs to make choices based on conditions, and master looping structures to create repetitive actions efficiently. By the end of this lab, students will be well-prepared to apply these fundamental concepts in Java programming, paving the way for more complex and exciting projects in the future.

Java is a versatile programming language that excels in the realm of parallel and distributed programming. Its robust support for multithreading and the Java Virtual Machine's (JVM) ability to manage threads efficiently make it a powerful choice for parallel computing. Java also offers libraries and frameworks like Java Concurrency Utilities and the `java.util.concurrent` package, which simplify the development of parallel applications.

Moreover, Java's platform independence allows developers to create distributed systems that can run seamlessly across different environments and devices, making it an ideal choice for building scalable and distributed applications, such as cloud services and large-scale data processing systems. Overall, Java's features and ecosystem make it a compelling option for tackling the complexities of parallel and distributed programming.

Advantages of java for parallel and distributed computing:

Java offers several advantages for parallel and distributed computing:

1. **Platform Independence:** Java's "Write Once, Run Anywhere" philosophy ensures that code can be developed on one platform and run on various others without modification. This portability is particularly advantageous in distributed computing, where different systems may be involved.
2. **Multithreading Support:** Java provides built-in support for multithreading, allowing developers to create concurrent applications easily. This is vital for exploiting parallelism in both parallel and distributed computing environments.
3. **Robustness and Reliability:** Java's strong typing, exception handling, and garbage collection contribute to creating reliable and robust applications. In distributed systems, where components may fail or encounter network issues, Java's error handling mechanisms help maintain system stability.
4. **Networking Capabilities:** Java includes extensive libraries for networking, enabling the development of distributed systems and communication between distributed components. The Java Remote Method Invocation (RMI) framework, for example, simplifies remote procedure calls in distributed applications.
5. **Security:** Java's security model, with features like the "sandboxing" of applets, makes it a trusted choice for distributed environments. It allows for fine-grained control over code execution and access to system resources.

6. **Scalability:** Java's support for building modular and reusable code components facilitates the development of scalable distributed systems. Components can be easily distributed across multiple machines and scaled up or down as needed.
7. **Community and Ecosystem:** Java boasts a vast and active developer community, resulting in a wealth of libraries, frameworks, and tools that support distributed and parallel computing. This ecosystem accelerates development and troubleshooting.
8. **Performance:** While Java's performance was once a concern, modern JVMs have made significant optimizations. With the right tuning and profiling, Java applications can achieve competitive performance levels in parallel and distributed computing scenarios.
9. **Enterprise-Ready:** Java is well-suited for enterprise-level applications, making it an ideal choice for building large-scale, distributed systems commonly found in business environments.
10. **Compatibility:** Older Java applications can often be seamlessly integrated into new distributed systems, extending the life and functionality of existing codebases.

Overall, Java's combination of portability, concurrency support, security features, and a robust ecosystem make it a compelling choice for parallel and distributed computing applications in various domains, including cloud computing, big data processing, and web services.

Activity-1:

A shop will give discount of 10% if the cost of purchased quantity is more than 1000.
Ask user for quantity. Suppose, one unit will cost 100. Judge and print total cost for user.

Solution:

Output Screenshot:

Activity-2:

Write a program to print absolute value of a number entered by user. E.g.-

INPUT: 1 OUTPUT: 1

INPUT: -1 OUTPUT: 1

Solution:

Output Screenshot:

Activity-3:

A school has following rules for grading system:

- a. Below 25 - F
- b. 25 to 45 - E
- c. 45 to 50 - D
- d. 50 to 60 - C
- e. 60 to 80 - B
- f. Above 80 - A

Ask user to enter marks and print the corresponding grade.

Solution:

Output Screenshot:

Activity-4:

Take values of length and breadth of a rectangle from user and check if it is square or not.

Solution:

Output Screenshot: