



# Java Basics

A Guide to Understanding the Fundamentals of Java Programming  
By: Dr. Tanvi


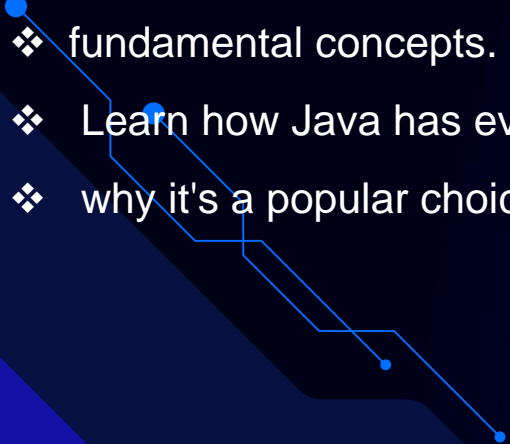


The background is a dark blue gradient with various geometric elements. There are several white and light blue lines, some straight and some forming a stepped path. Small blue squares and circles are scattered throughout. In the top right and bottom left corners, there are patterns of parallel diagonal lines in a lighter blue shade.

# 1. INTRODUCTION



# Introduction

■ This presentation covers the basics of Java programming,

- ❖ introducing its history,
  - ❖ key features, and
  - ❖ fundamental concepts.
  - ❖ Learn how Java has evolved and
  - ❖ why it's a popular choice for developers.
- 
- 
- 
- 

# Overview of Java History

Java was created by Sun Microsystems in the mid-1990s as a versatile programming language. Its design focuses on simplicity, security, and portability. Initially developed for interactive television, it gained popularity for web and application development, leading to its influence across industries

## Early development

- 1991**: James Gosling, Mike Sheridan, and Patrick Naughton start the Green Project to create a new programming language
- 1995**: The first version of Java, called Oak, is designed for consumer electronics
- 1996**: The first official release of Java, JDK 1.0, is released

## Later versions

- 2004**: Java SE 5.0 introduces generics and enhanced "for" loops
- 2014**: Java SE 8 introduces lambda expressions and the Stream API
- 2017**: Java SE 9 introduces the module system and JShell (REPL)
- 2018**: Java SE 11 is the first long-term support (LTS) release
- 2021**: Java SE 17, another LTS release, introduces sealed classes

# Key Features of Java

Platform independent

Robustness

Automatic garbage collection

Rich API

Strong memory management

Built-in-security

Reliability

Multithreading

High performance

Secure

# Java Development Environment

Setting up a Java development environment involves

- ❖ installing the Java Development Kit (JDK), which includes tools for compiling and running Java programs.
- ❖ An Integrated Development Environment (IDE) like Eclipse or IntelliJ IDEA is often used to enhance productivity with features like code completion, debugging, and project management.
- ❖ Configuration of environment variables is also necessary for the seamless operation of Java applications.

# Architecture of JAVA

(JDK) Java Development Kit

JRE(Java Runtime Environment)

JVM

Class  
Loader

Memory  
Area

Execution Engine

Java  
Packages  
Classes

Runtime  
Libraries

Development Tools

.java  
.javac  
.jar  
.javadoc  
.jdb  
.etc..



02

# Java Basics

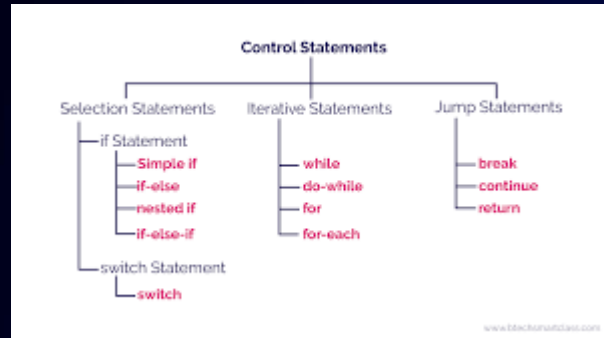


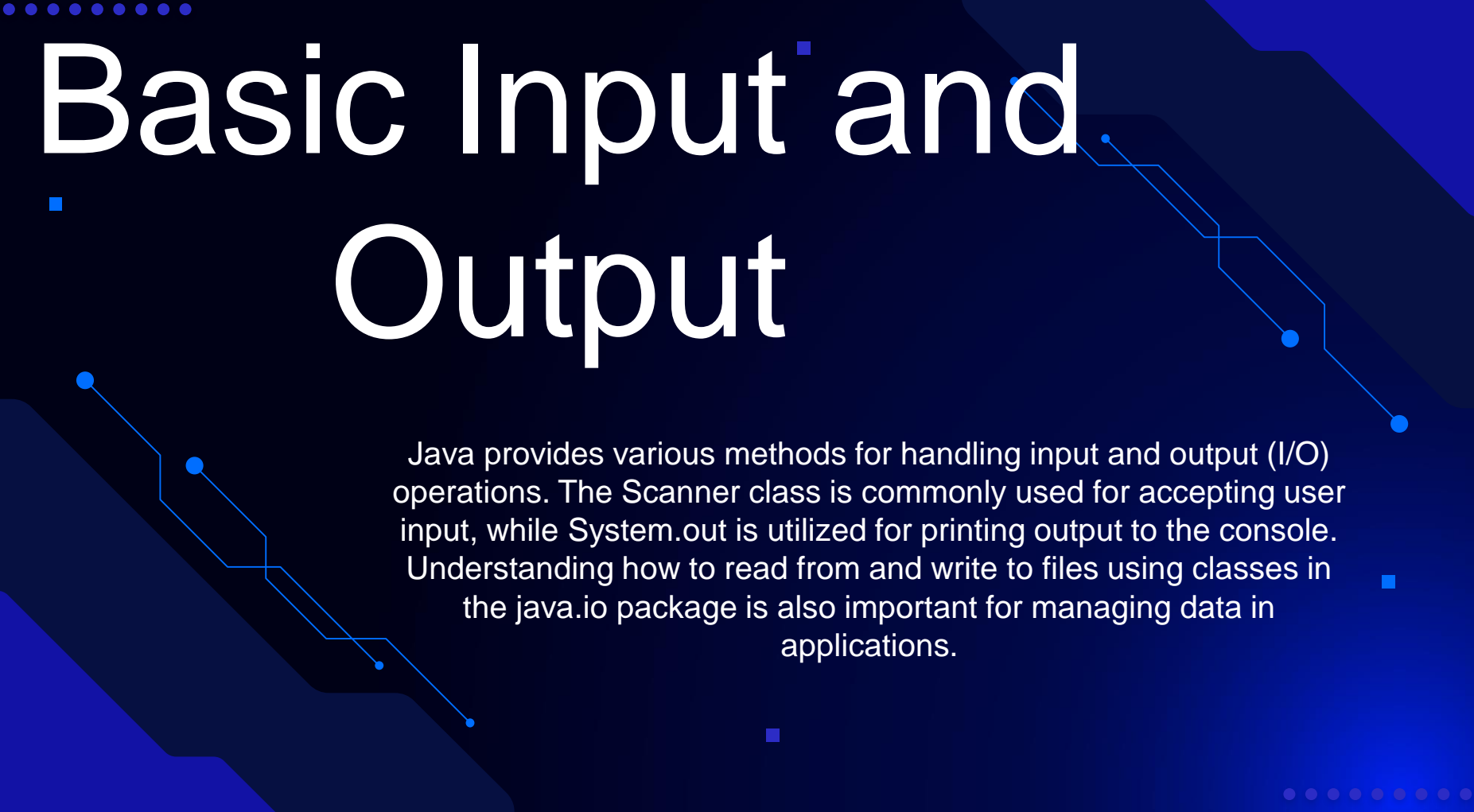
# Variables and Data Types

Variables in Java are used to store data values and can be defined with specific data types such as int, double, char, and String. Each data type has its own characteristics and storage requirements. For example, int stores integers, while String is used for sequences of characters. Understanding how to declare and initialize variables is crucial for efficient Java programming.

# Control Structures

Control structures in Java dictate the flow of execution of the program. Common structures include conditional statements like if-else and switch, which allow for decision-making in code. Loops such as for, while, and do-while enable repeated execution of code blocks. Mastering these control structures is key to building robust Java applications.



The slide features a dark blue background with abstract geometric shapes and decorative elements. At the top left, there is a horizontal row of seven small blue dots. A series of blue lines with circular nodes at the ends, resembling a circuit or data path, starts from the top right and extends towards the bottom left. Another similar line starts from the bottom left and extends towards the top right. Small blue squares are scattered throughout the slide, including one near the top center and another near the bottom center.

# Basic Input and Output

Java provides various methods for handling input and output (I/O) operations. The Scanner class is commonly used for accepting user input, while System.out is utilized for printing output to the console. Understanding how to read from and write to files using classes in the java.io package is also important for managing data in applications.

# Conclusions

In summary, Java is a powerful programming language with a rich set of features that simplify development. Understanding the development environment, basic syntax, data types, control structures, and I/O operations lays the groundwork for any aspiring Java programmer. With these fundamentals, developers can create a wide range of applications, from simple to complex.