Name: _____

Java Scanner

What is Scanner?

Scanner is a Java class that lets your program read input from the user. It's like giving your program "ears" to listen to what the user types on the keyboard. Without Scanner, your program can only talk (output) but can't listen (input).

Simple Analogy: Scanner is Like Being a Good Listener

Think of Scanner like being a careful listener in a conversation:

- **You put on headphones** = `Scanner input = new Scanner(System.in);`
  - You get ready to listen carefully
- **You listen for different things** = `nextInt()`, `nextLine()`, `nextDouble()`
  - Sometimes you listen for numbers, sometimes for full sentences
- ☐ **You wait patiently** = Program pauses until user types something
  - You don't interrupt - you wait for them to finish
- **You remember what they said** = Store input in variables
  - You save important information to use later

Just like being a good listener, Scanner waits patiently and captures exactly what the user says!

Scanner Setup (ALWAYS Required!)

```
// Step 1: At the very top of your program file
import java.util.Scanner;

// Step 2: Inside your main method, create a Scanner
Scanner input = new Scanner(System.in);

// Step 3: Use it to get input (see examples below)

// Step 4: (Optional) Close it when done

input.close();
```

Remember: Create your Scanner ONCE at the beginning, then use it throughout your entire program!

Scanner Methods: Different Ways to Listen

Method	What It Reads	Example Input	Stops Reading At
nextLine()	Entire line of text (including spaces)	"Hello World 123"	Enter key
next()	Single word (no spaces)	"Hello" (ignores " World 123")	Space or Enter
nextInt()	Integer number only	25	Space or Enter
nextDouble()	Decimal number only	3.14	Space or Enter
nextBoolean()	true or false only	true	Space or Enter

Example 1: Basic Input Collection

```java
import java.util.Scanner;
public class UserInfo {
 public static void main(String[] args) {
   Scanner input = new Scanner(System.in);

    // Get different types of information
    System.out.print("What's your full name? ");
    String fullName = input.nextLine();

    System.out.print("How old are you? ");
    int age = input.nextInt();
    System.out.print("What's your height in feet? ");
    double height = input.nextDouble();
    // Display the collected information
    System.out.println("\n--- Your Information ---");
    System.out.println("Name: " + fullName);
    System.out.println("Age: " + age + " years old");
    System.out.println("Height: " + height + " feet");

    input.close();
  }
}
```

What user sees and types:

```
What's your full name? Maria Garcia
How old are you? 16
What's your height in feet? 5.4

--- Your Information ---
Name: Maria Garcia
Age: 16 years old
Height: 5.4 feet
```

Example 2: Building a Simple Calculator

```java
Scanner input = new Scanner(System.in);
System.out.println("=== Simple Calculator ===");
System.out.print("Enter first number: ");
double num1 = input.nextDouble();

System.out.print("Enter second number: ");
double num2 = input.nextDouble();

// Perform calculations
double sum = num1 + num2;
double difference = num1 - num2;
double product = num1 * num2;
double quotient = num1 / num2;

System.out.println("\n--- Results ---");
System.out.println(num1 + " + " + num2 + " = " + sum);
System.out.println(num1 + " - " + num2 + " = " + difference);
System.out.println(num1 + " × " + num2 + " = " + product);
System.out.println(num1 + " ÷ " + num2 + " = " + quotient);
```

⚠ The Most Common Scanner Problem: Mixing nextInt() with nextLine()

The Problem:

```
// This code has a BUG!
System.out.print("Enter your age: ");
int age = input.nextInt(); // Reads number, leaves Enter key behind
System.out.print("Enter your name: ");
String name = input.nextLine(); // Gets empty string!
```

What happens: After typing the age and pressing Enter, **`nextInt()`** reads the number but leaves the Enter key press in the "buffer." When **`nextLine()`** runs, it immediately reads that leftover Enter key and thinks the user entered an empty line!

The Solution:

```
// FIXED VERSION
System.out.print("Enter your age: ");
int age = input.nextInt();
input.nextLine(); // Clear the leftover Enter key
System.out.print("Enter your name: ");
String name = input.nextLine(); // Now this works correctly!
```

Rule: Always add **`input.nextLine();`** after using **`nextInt()`**, **`nextDouble()`**, etc., if you plan to use **`nextLine()`** afterward.

Example 3: Interactive Menu System

```
Scanner input = new Scanner(System.in);
boolean keepRunning = true;
while (keepRunning) {
  System.out.println("\n=== MAIN MENU ===");
  System.out.println("1. Say Hello");
  System.out.println("2. Calculate Square");
  System.out.println("3. Exit");
  System.out.print("Choose an option (1-3): ");
  int choice = input.nextInt();
  if (choice == 1) {
    System.out.print("What's your name? ");
    input.nextLine(); // Clear buffer
    String name = input.nextLine();
    System.out.println("Hello, " + name + "!");
  } else if (choice == 2) {
    System.out.print("Enter a number: ");
    double number = input.nextDouble();
    double square = number * number;
    System.out.println(number + " squared is " + square);
  } else if (choice == 3) {
    System.out.println("Goodbye!");
    keepRunning = false;
  } else {
    System.out.println("Invalid option. Please try again.");
  }
}
```

Example 4: Input Validation

```
Scanner input = new Scanner(System.in);
// Keep asking until we get a valid age
int age = -1;
while (age < 0 || age > 120) {
  System.out.print("Enter your age (0-120): ");

  if (input.hasNextInt()) { // Check if next input is an integer
    age = input.nextInt();
    if (age < 0 || age > 120) {
      System.out.println("Age must be between 0 and 120!");
    }
  } else {
    System.out.println("Please enter a valid number!");
    input.next(); // Skip the invalid input
  }
}
System.out.println("Valid age entered: " + age);
```

Example 5: Processing Multiple Items

```
Scanner input = new Scanner(System.in);

System.out.print("How many numbers do you want to enter? ");
int count = input.nextInt();

double sum = 0;
for (int i = 1; i <= count; i++) {
  System.out.print("Enter number " + i + ": ");
  double number = input.nextDouble();
  sum += number;
}
double average = sum / count;
System.out.println("Average: " + average);
```

```
How many numbers do you want to enter? 3
Enter number 1: 10.5
Enter number 2: 8.0
Enter number 3: 9.5
Average: 9.33
```

Scanner Best Practices

Good Practices

- **Import Scanner** at the top of your file
- **Create Scanner once** at the beginning
- **Use clear prompts:** "Enter your age: "
- **Add space** after prompts with print(), not println()
- **Validate input** when necessary
- **Close Scanner** when completely done
- **Use nextLine()** for text with spaces

Common Mistakes

- **Forgetting import** statement
- **Creating multiple Scanners** unnecessarily
- **Vague prompts:** "Enter:", "Input:"
- **Missing spaces** in prompts
- **Not handling errors** gracefully
- **Mixing nextInt() with nextLine()** without clearing buffer
- **Using next()** when you want full sentences

Real-World Example: Grade Calculator

```
import java.util.Scanner;
public class GradeCalculator {
  public static void main(String[] args) {
    Scanner input = new Scanner(System.in);

    System.out.println("=== Grade Calculator ===");
    System.out.print("Enter student's name: ");
    String studentName = input.nextLine();
    System.out.print("How many assignments? ");
    int numAssignments = input.nextInt();
    double totalPoints = 0;
    double maxPoints = 0;
    for (int i = 1; i <= numAssignments; i++) {
      System.out.print("Assignment " + i + " points earned: ");
      double earned = input.nextDouble();

      System.out.print("Assignment " + i + " points possible: ");
      double possible = input.nextDouble();
      totalPoints += earned;
      maxPoints += possible;
    }
    double percentage = (totalPoints / maxPoints) * 100;
    String letterGrade;
    if (percentage >= 90) {
      letterGrade = "A";
    } else if (percentage >= 80) {
      letterGrade = "B";
    } else if (percentage >= 70) {
      letterGrade = "C";
    } else if (percentage >= 60) {
      letterGrade = "D";
    } else {
      letterGrade = "F";
    }
    System.out.println("\n=== Grade Report ===");
    System.out.println("Student: " + studentName + "\n");
    System.out.println("Points: " + totalPoints + " / " + maxPoints);
    System.out.println("Percentage: " + percentage + "%");
    System.out.println("Letter Grade: " + letterGrade);

    input.close();
  }
}
```

Troubleshooting Scanner Issues

Problem	Symptom	Solution
Buffer not cleared	nextLine() gets empty string	Add input.nextLine() after nextInt()
Wrong input type	Program crashes	Use hasNextInt() to check first
Spaces in input	next() only reads first word	Use nextLine() instead
Scanner not imported	Compile error	Add import java.util.Scanner;

Your Turn: Write Your Own Definition

What is Scanner in Java? How would you explain it to a friend?

Write your definition in your own words:

Match each Scanner method with the best use case:

Method	Best Used For
nextLine()	_____
nextInt()	_____
nextDouble()	_____
next()	_____

Why do you think it's important to validate user input?

Describe a program you could create that would use Scanner. What would it ask the user?