Simple Stopwatch:

Input: Start and stop commands.
Output: Elapsed time when the stopwatch is stopped.

Example:

- Input: "Start stopwatch"
- Output: "Stopwatch started."
- After some time: "Stop stopwatch"
- Output: "Elapsed time: 12.34 seconds."

Solution 1: Simple Stopwatch using System.currentTimeMillis()

Code:

```
import java.util.Scanner;

public class Stopwatch {
    // Variables to store start and stop times
    private long startTime;
    private long stopTime;

    // Method to start the stopwatch
    public void start() {
        startTime = System.currentTimeMillis();  // Capture the start time
        System.out.println("Stopwatch started.");
    }

    // Method to stop the stopwatch and calculate elapsed time
    public void stop() {
        stopTime = System.currentTimeMillis();  // Capture the stop time
        long elapsedTime = stopTime - startTime;  // Calculate the elapsed
        System.out.println("Elapsed time: " + (elapsedTime / 1000.0) + " se
    }

    // Main method to run the stopwatch
    public static void main(String[] args) {
        Stopwatch stopwatch = new Stopwatch();
        Scanner scanner = new Scanner(System.in);
```

```
            System.out.println("Enter 'start' to begin the stopwatch:");
            String input = scanner.nextLine();

            if (input.equalsIgnoreCase("start")) {
                stopwatch.start();  // Start the stopwatch
            }

            System.out.println("Enter 'stop' to stop the stopwatch:");
            input = scanner.nextLine();

            if (input.equalsIgnoreCase("stop")) {
                stopwatch.stop();  // Stop the stopwatch
            }

            scanner.close();
        }
    }
```

Output:

```
Enter 'start' to begin the stopwatch:
 start
Stopwatch started.
```

```
Enter 'stop' to stop the stopwatch:
 stop
Elapsed time: 8.425 seconds.
```

Explanation :

- Uses System.currentTimeMillis() to record time in milliseconds.
- Starts and stops the stopwatch using user input.
- Calculates and prints the elapsed time in seconds.

Solution 2: Simple Stopwatch using nanoTime()

Code:

```
    import java.util.Scanner;                                    Copy

    public class NanoStopwatch {
```

```
    // Variables to store start and stop times
    private long startTime;
    private long stopTime;

    // Method to start the stopwatch
    public void start() {
        startTime = System.nanoTime();  // Capture the start time in nanose
        System.out.println("Stopwatch started.");
    }

    // Method to stop the stopwatch and calculate elapsed time
    public void stop() {
        stopTime = System.nanoTime();  // Capture the stop time in nanosec
        long elapsedTime = stopTime - startTime;  // Calculate the elapsed
        System.out.println("Elapsed time: " + (elapsedTime / 1_000_000_000
    }

    // Main method to run the stopwatch
    public static void main(String[] args) {
        NanoStopwatch stopwatch = new NanoStopwatch();
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter 'start' to begin the stopwatch:");
        String input = scanner.nextLine();

        if (input.equalsIgnoreCase("start")) {
            stopwatch.start();  // Start the stopwatch
        }

        System.out.println("Enter 'stop' to stop the stopwatch:");
        input = scanner.nextLine();

        if (input.equalsIgnoreCase("stop")) {
            stopwatch.stop();  // Stop the stopwatch
        }

        scanner.close();
    }
}
```

Output:

```
Enter 'start' to begin the stopwatch:
 start
Stopwatch started.
```

```
Enter 'stop' to stop the stopwatch:
 stop
Elapsed time: 16.259088781 seconds.
```

Explanation:

- Uses System.nanoTime() for higher precision timing in nanoseconds.
- Same user input logic as the first solution, but with greater accuracy.
- Outputs the time in seconds after converting nanoseconds.

Java Code Editor: