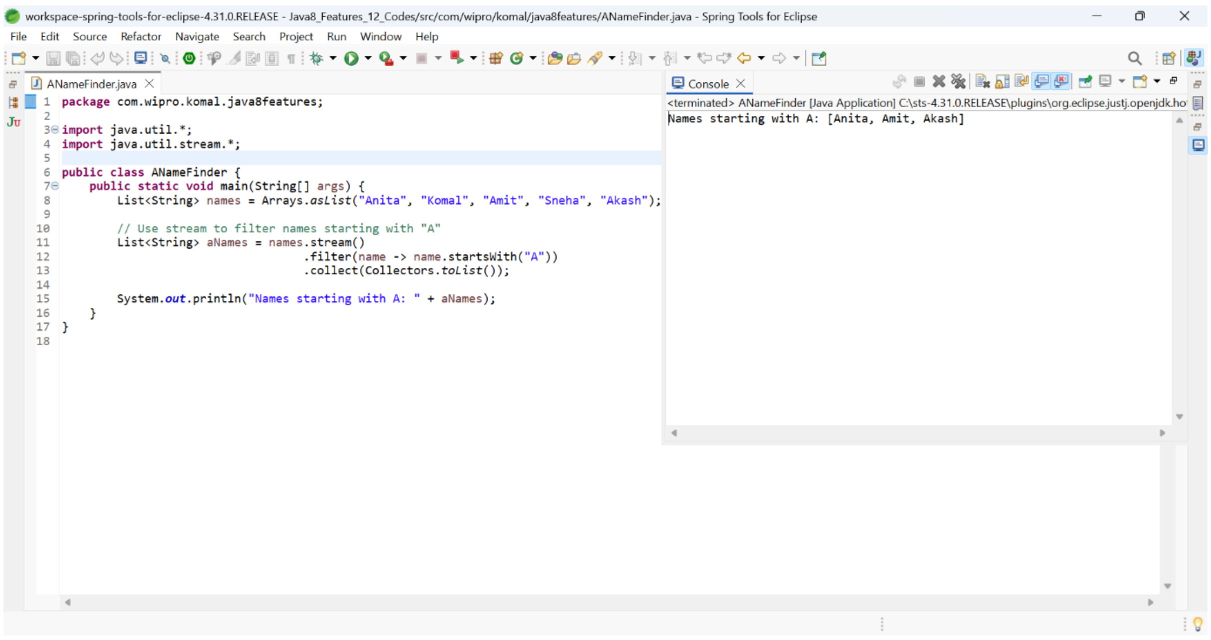
Name:- Komal Sudhakar Baviskar

#Java_8_features codes Output

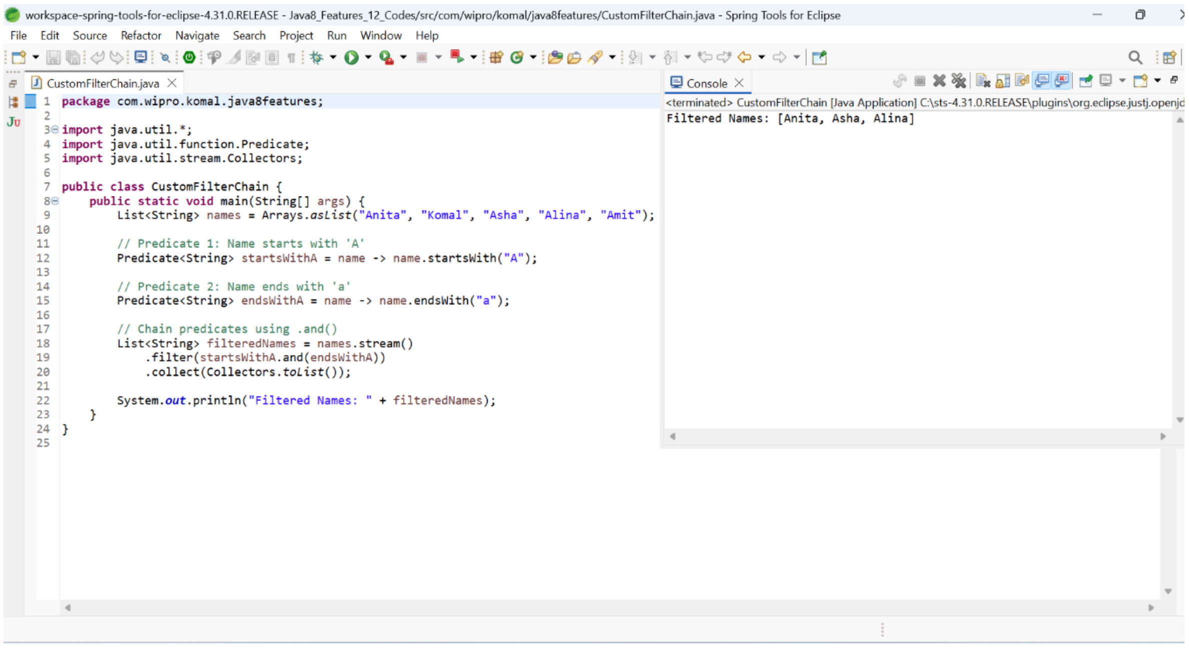
1)


workspace-spring-tools-for-eclipse-4.31.0.RELEASE - Java8_Features_12_Codes/src/com/wipro/komal/java8features/ANameFinder.java - Spring Tools for Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
ANameFinder.java
Console

```java
package com.wipro.komal.java8features;

import java.util.*;
import java.util.stream.*;

public class ANameFinder {
    public static void main(String[] args) {
        List<String> names = Arrays.asList("Anita", "Komal", "Amit", "Sneha", "Akash");

        // Use stream to filter names starting with "A"
        List<String> aNames = names.stream()
                                   .filter(name -> name.startsWith("A"))
                                   .collect(Collectors.toList());

        System.out.println("Names starting with A: " + aNames);
    }
}
```

```
<terminated> ANameFinder [Java Application] C:\sts-4.31.0.RELEASE\plugins\org.eclipse.justj.openjdk.ho
Names starting with A: [Anita, Amit, Akash]
```

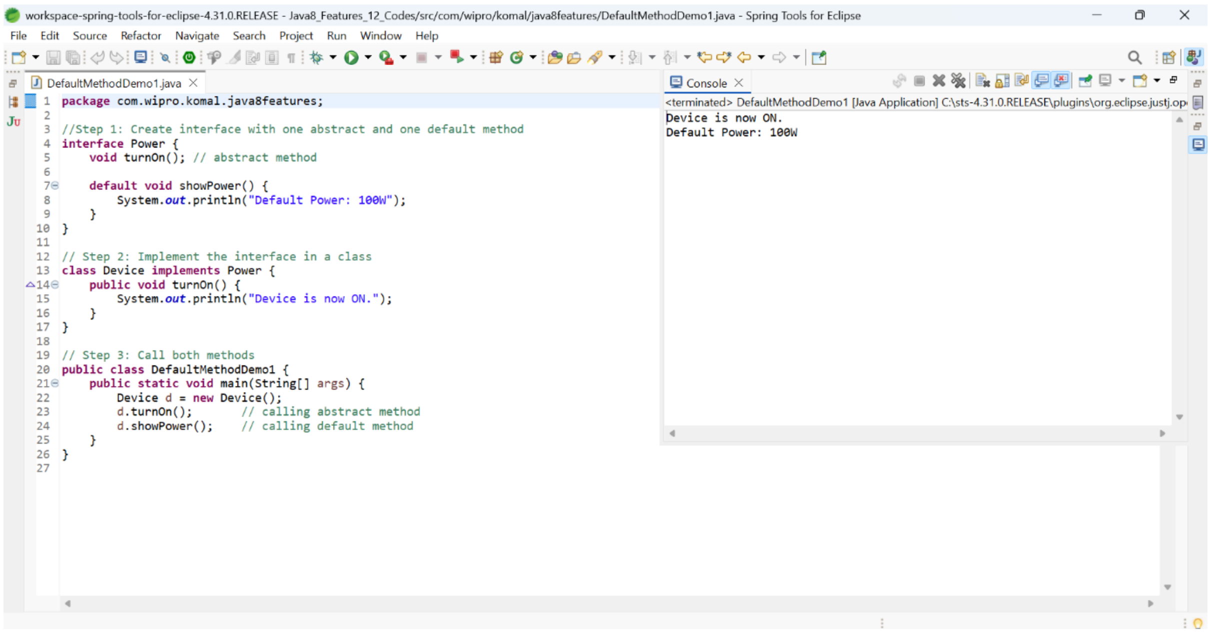
2)


workspace-spring-tools-for-eclipse-4.31.0.RELEASE - Java8_Features_12_Codes/src/com/wipro/komal/java8features/CustomFilterChain.java - Spring Tools for Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
CustomFilterChain.java
Console

```java
package com.wipro.komal.java8features;

import java.util.*;
import java.util.function.Predicate;
import java.util.stream.Collectors;

public class CustomFilterChain {
    public static void main(String[] args) {
        List<String> names = Arrays.asList("Anita", "Komal", "Asha", "Alina", "Amit");

        // Predicate 1: Name starts with 'A'
        Predicate<String> startsWithA = name -> name.startsWith("A");

        // Predicate 2: Name ends with 'a'
        Predicate<String> endsWithA = name -> name.endsWith("a");

        // Chain predicates using .and()
        List<String> filteredNames = names.stream()
            .filter(startsWithA.and(endsWithA))
            .collect(Collectors.toList());

        System.out.println("Filtered Names: " + filteredNames);
    }
}
```

```
<terminated> CustomFilterChain [Java Application] C:\sts-4.31.0.RELEASE\plugins\org.eclipse.justj.openjd
Filtered Names: [Anita, Asha, Alina]
```

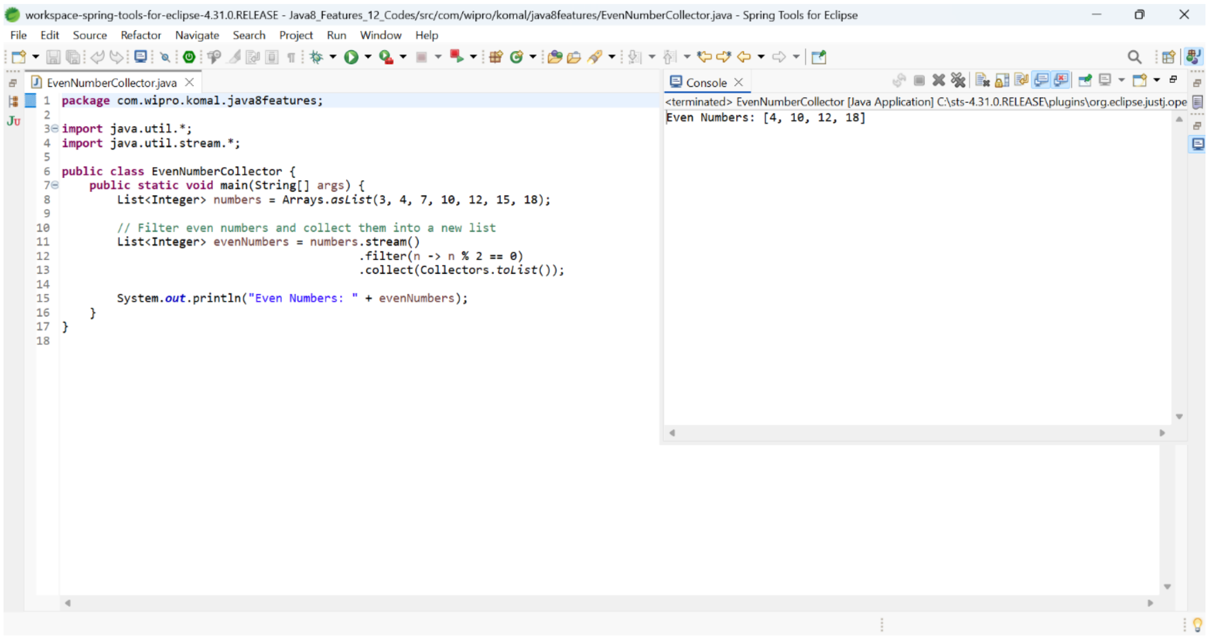
3)


workspace-spring-tools-for-eclipse-4.31.0.RELEASE - Java8_Features_12_Codes/src/com/wipro/komal/java8features/DefaultMethodDemo1.java - Spring Tools for Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
DefaultMethodDemo1.java
Console

```java
package com.wipro.komal.java8features;

//Step 1: Create interface with one abstract and one default method
interface Power {
    void turnOn(); // abstract method

    default void showPower() {
        System.out.println("Default Power: 100W");
    }
}

// Step 2: Implement the interface in a class
class Device implements Power {
    public void turnOn() {
        System.out.println("Device is now ON.");
    }
}

// Step 3: Call both methods
public class DefaultMethodDemo1 {
    public static void main(String[] args) {
        Device d = new Device();
        d.turnOn();        // calling abstract method
        d.showPower();     // calling default method
    }
}
```

```
<terminated> DefaultMethodDemo1 [Java Application] C:\sts-4.31.0.RELEASE\plugins\org.eclipse.justj.op
Device is now ON.
Default Power: 100W
```

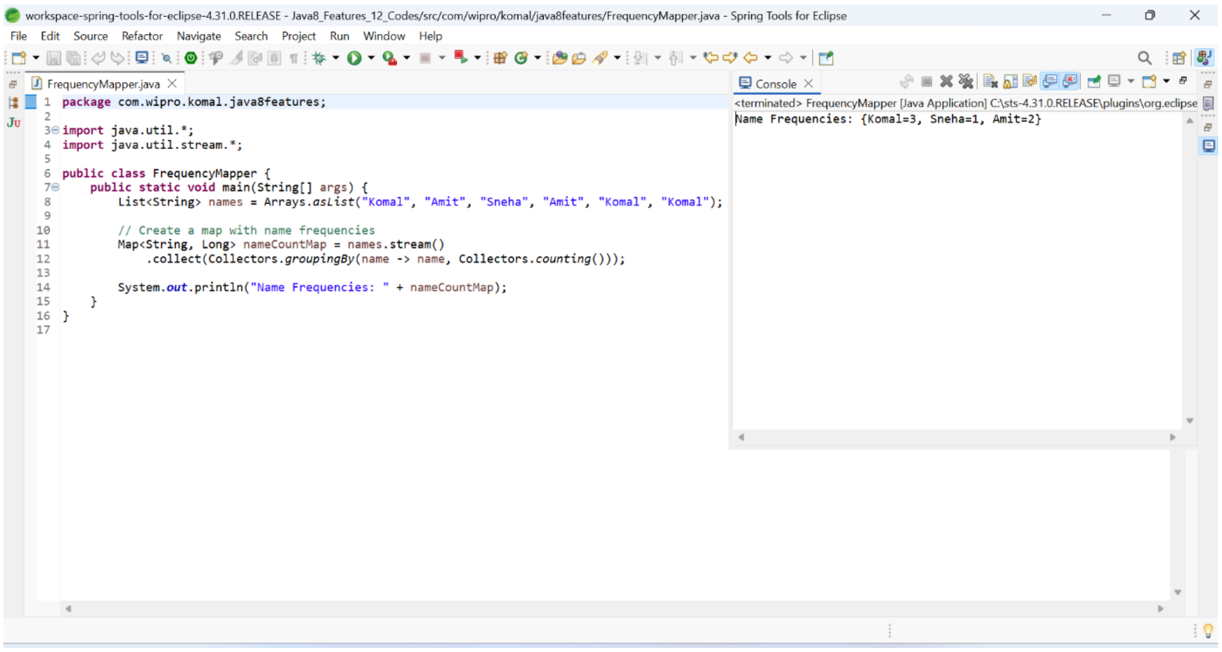
4)


workspace-spring-tools-for-eclipse-4.31.0.RELEASE - Java8_Features_12_Codes/src/com/wipro/komal/java8features/EvenNumberCollector.java - Spring Tools for Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
EvenNumberCollector.java
Console

```java
package com.wipro.komal.java8features;

import java.util.*;
import java.util.stream.*;

public class EvenNumberCollector {
    public static void main(String[] args) {
        List<Integer> numbers = Arrays.asList(3, 4, 7, 10, 12, 15, 18);

        // Filter even numbers and collect them into a new list
        List<Integer> evenNumbers = numbers.stream()
                                   .filter(n -> n % 2 == 0)
                                   .collect(Collectors.toList());

        System.out.println("Even Numbers: " + evenNumbers);
    }
}
```

```
<terminated> EvenNumberCollector [Java Application] C:\sts-4.31.0.RELEASE\plugins\org.eclipse.justj.ope
Even Numbers: [4, 10, 12, 18]
```

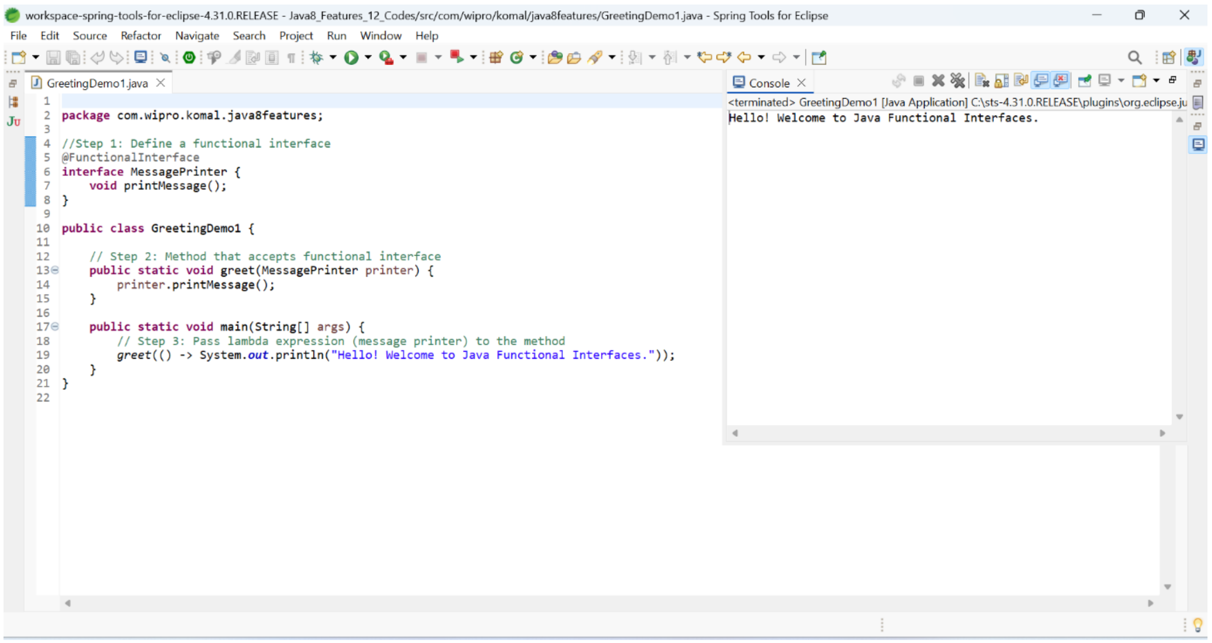
5)


workspace-spring-tools-for-eclipse-4.31.0.RELEASE - Java8_Features_12_Codes/src/com/wipro/komal/java8features/FrequencyMapper.java - Spring Tools for Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
FrequencyMapper.java
Console
<terminated> FrequencyMapper [Java Application] C:\sts-4.31.0.RELEASE\plugins\org.eclipse

```java
package com.wipro.komal.java8features;

import java.util.*;
import java.util.stream.*;

public class FrequencyMapper {
    public static void main(String[] args) {
        List<String> names = Arrays.asList("Komal", "Amit", "Sneha", "Amit", "Komal", "Komal");

        // Create a map with name frequencies
        Map<String, Long> nameCountMap = names.stream()
            .collect(Collectors.groupingBy(name -> name, Collectors.counting()));

        System.out.println("Name Frequencies: " + nameCountMap);
    }
}
```

```
Name Frequencies: {Komal=3, Sneha=1, Amit=2}
```

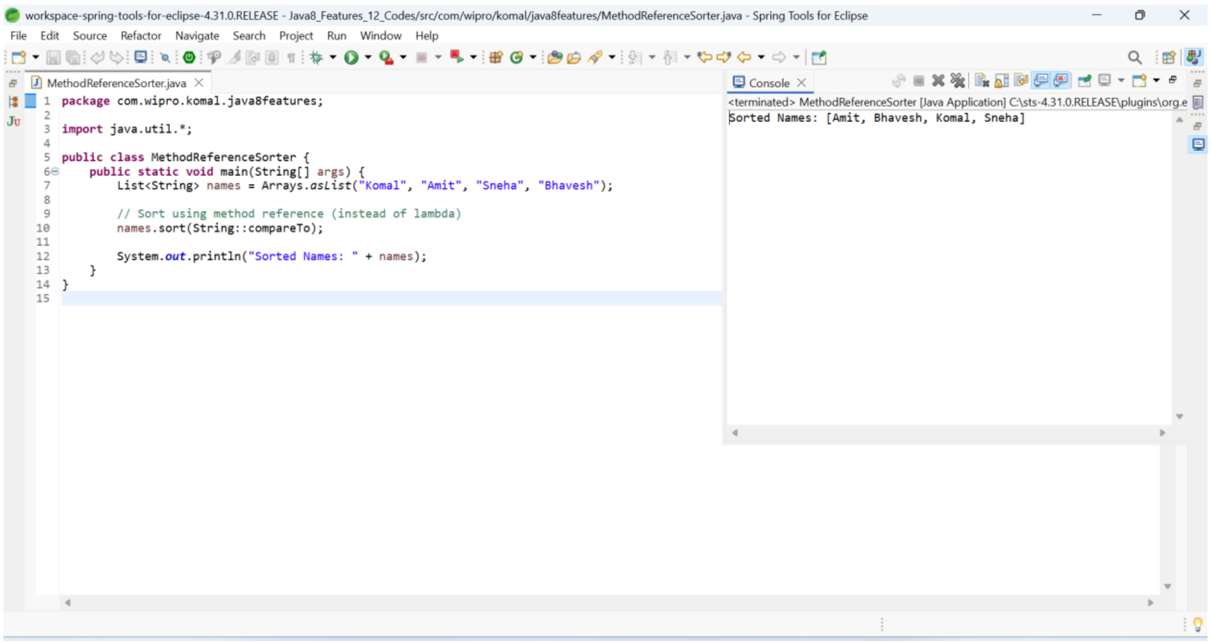
6)


workspace-spring-tools-for-eclipse-4.31.0.RELEASE - Java8_Features_12_Codes/src/com/wipro/komal/java8features/GreetingDemo1.java - Spring Tools for Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
GreetingDemo1.java
Console
<terminated> GreetingDemo1 [Java Application] C:\sts-4.31.0.RELEASE\plugins\org.eclipse.ju

```java

package com.wipro.komal.java8features;

//Step 1: Define a functional interface
@FunctionalInterface
interface MessagePrinter {
    void printMessage();
}

public class GreetingDemo1 {

    // Step 2: Method that accepts functional interface
    public static void greet(MessagePrinter printer) {
        printer.printMessage();
    }

    public static void main(String[] args) {
        // Step 3: Pass lambda expression (message printer) to the method
        greet(() -> System.out.println("Hello! Welcome to Java Functional Interfaces."));
    }
}
```

```
Hello! Welcome to Java Functional Interfaces.
```

7)

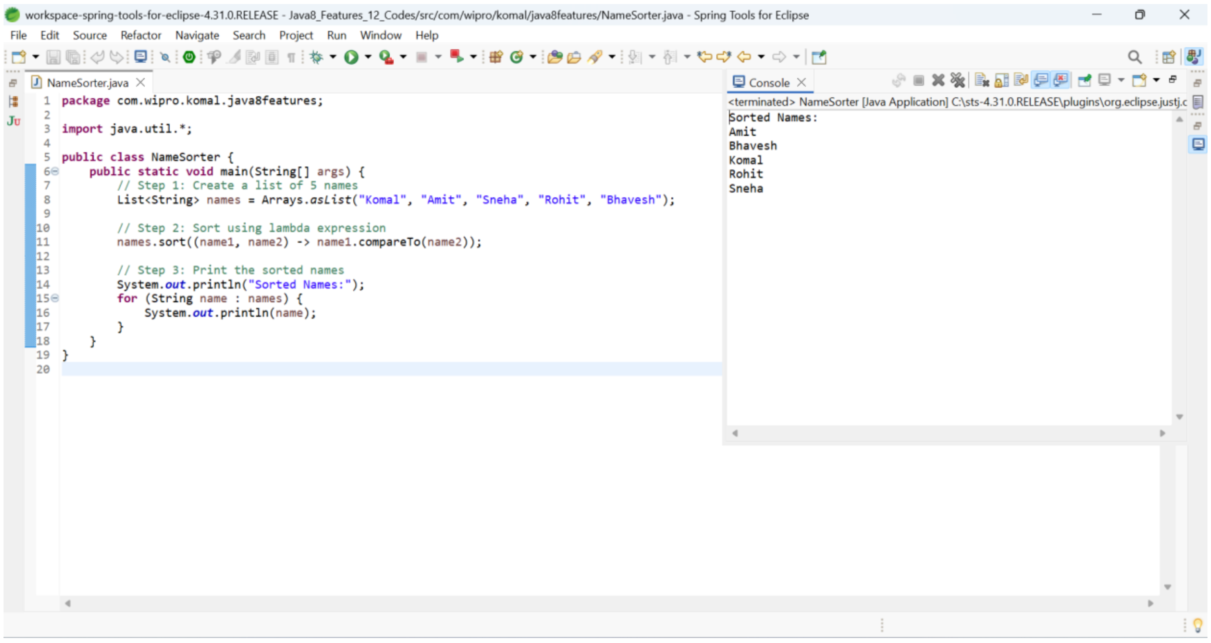


workspace-spring-tools-for-eclipse-4.31.0.RELEASE - Java8_Features_12_Codes/src/com/wipro/komal/java8features/MethodReferenceSorter.java - Spring Tools for Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
MethodReferenceSorter.java
Console

```java
package com.wipro.komal.java8features;

import java.util.*;

public class MethodReferenceSorter {
    public static void main(String[] args) {
        List<String> names = Arrays.asList("Komal", "Amit", "Sneha", "Bhavesh");

        // Sort using method reference (instead of lambda)
        names.sort(String::compareTo);

        System.out.println("Sorted Names: " + names);
    }
}
```

```
<terminated> MethodReferenceSorter [Java Application] C:\sts-4.31.0.RELEASE\plugins\org.e
Sorted Names: [Amit, Bhavesh, Komal, Sneha]
```

8)

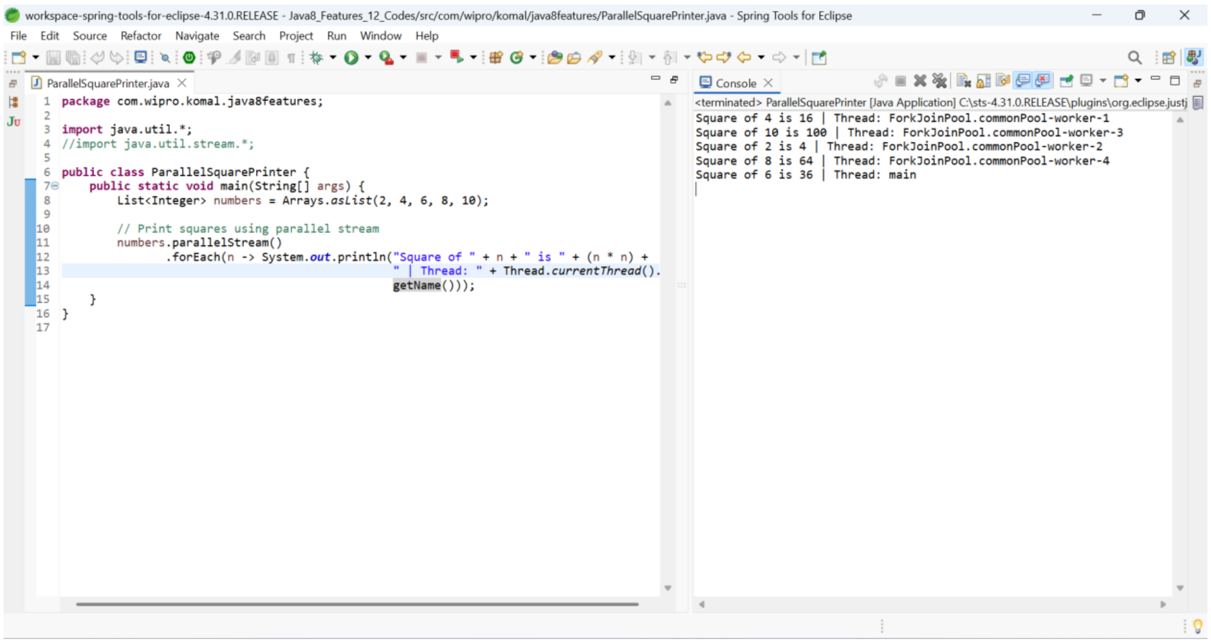


workspace-spring-tools-for-eclipse-4.31.0.RELEASE - Java8_Features_12_Codes/src/com/wipro/komal/java8features/NameSorter.java - Spring Tools for Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
NameSorter.java
Console

```java
package com.wipro.komal.java8features;

import java.util.*;

public class NameSorter {
    public static void main(String[] args) {
        // Step 1: Create a list of 5 names
        List<String> names = Arrays.asList("Komal", "Amit", "Sneha", "Rohit", "Bhavesh");

        // Step 2: Sort using lambda expression
        names.sort((name1, name2) -> name1.compareTo(name2));

        // Step 3: Print the sorted names
        System.out.println("Sorted Names:");
        for (String name : names) {
            System.out.println(name);
        }
    }
}
```

```
<terminated> NameSorter [Java Application] C:\sts-4.31.0.RELEASE\plugins\org.eclipse.justj.c
Sorted Names:
Amit
Bhavesh
Komal
Rohit
Sneha
```

9)



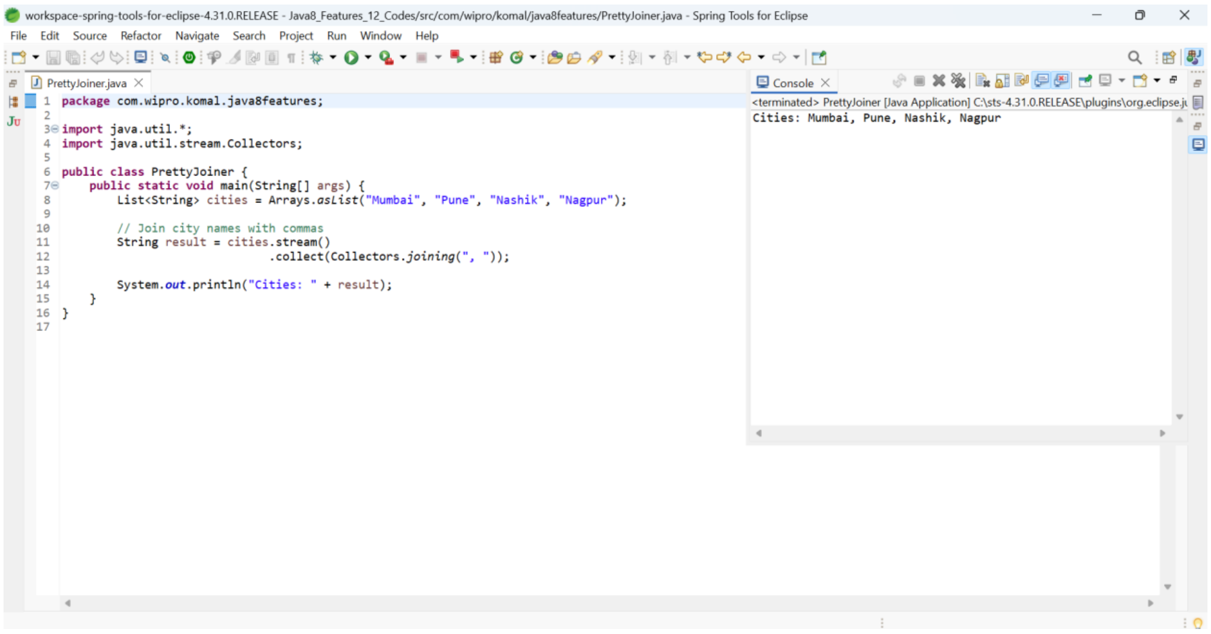
workspace-spring-tools-for-eclipse-4.31.0.RELEASE - Java8_Features_12_Codes/src/com/wipro/komal/java8features/ParallelSquarePrinter.java - Spring Tools for Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help
ParallelSquarePrinter.java
Console

```java
package com.wipro.komal.java8features;

import java.util.*;
//import java.util.stream.*;

public class ParallelSquarePrinter {
    public static void main(String[] args) {
        List<Integer> numbers = Arrays.asList(2, 4, 6, 8, 10);

        // Print squares using parallel stream
        numbers.parallelStream()
               .forEach(n -> System.out.println("Square of " + n + " is " + (n * n) +
                                                  " | Thread: " + Thread.currentThread().
                                                  getName()));

    }
}
```

```
<terminated> ParallelSquarePrinter [Java Application] C:\sts-4.31.0.RELEASE\plugins\org.eclipse.justj
Square of 4 is 16 | Thread: ForkJoinPool.commonPool-worker-1
Square of 10 is 100 | Thread: ForkJoinPool.commonPool-worker-3
Square of 2 is 4 | Thread: ForkJoinPool.commonPool-worker-2
Square of 8 is 64 | Thread: ForkJoinPool.commonPool-worker-4
Square of 6 is 36 | Thread: main
```

10)



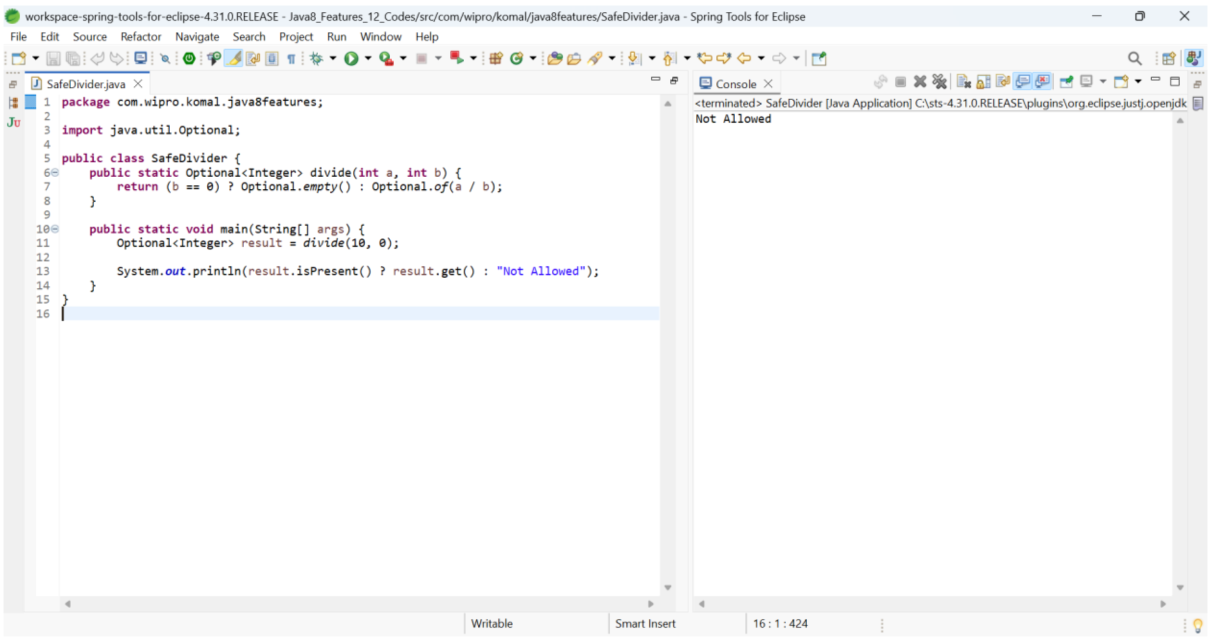
workspace-spring-tools-for-eclipse-4.31.0.RELEASE - Java8_Features_12_Codes/src/com/wipro/komal/java8features/PrettyJoiner.java - Spring Tools for Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help
PrettyJoiner.java
Console

```java
package com.wipro.komal.java8features;

import java.util.*;
import java.util.stream.Collectors;

public class PrettyJoiner {
    public static void main(String[] args) {
        List<String> cities = Arrays.asList("Mumbai", "Pune", "Nashik", "Nagpur");

        // Join city names with commas
        String result = cities.stream()
                              .collect(Collectors.joining(", "));

        System.out.println("Cities: " + result);
    }
}
```

```
<terminated> PrettyJoiner [Java Application] C:\sts-4.31.0.RELEASE\plugins\org.eclipse.ju
Cities: Mumbai, Pune, Nashik, Nagpur
```

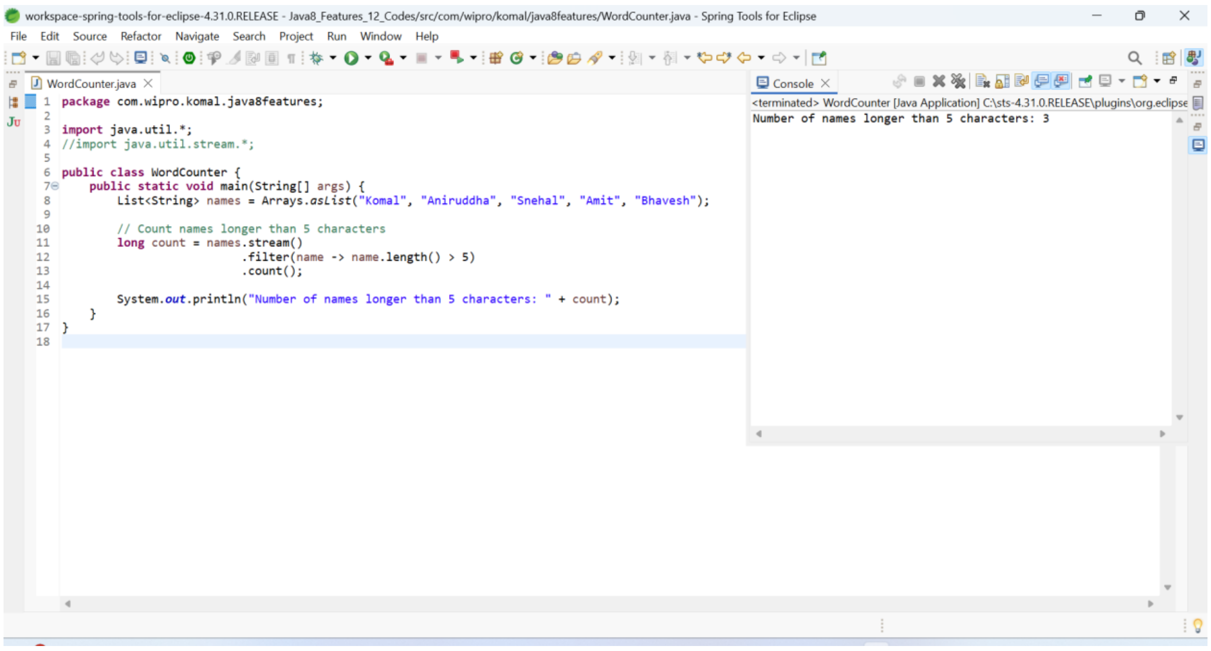
11)


workspace-spring-tools-for-eclipse-4.31.0.RELEASE - Java8_Features_12_Codes/src/com/wipro/komal/java8features/SafeDivider.java - Spring Tools for Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
SafeDivider.java
Console
Writable
Smart Insert
16 : 1 : 424

```java
package com.wipro.komal.java8features;

import java.util.Optional;

public class SafeDivider {
    public static Optional<Integer> divide(int a, int b) {
        return (b == 0) ? Optional.empty() : Optional.of(a / b);
    }

    public static void main(String[] args) {
        Optional<Integer> result = divide(10, 0);

        System.out.println(result.isPresent() ? result.get() : "Not Allowed");
    }
}
```

```
<terminated> SafeDivider [Java Application] C:\sts-4.31.0.RELEASE\plugins\org.eclipse.justj.openjdk
Not Allowed
```

12)


workspace-spring-tools-for-eclipse-4.31.0.RELEASE - Java8_Features_12_Codes/src/com/wipro/komal/java8features/WordCounter.java - Spring Tools for Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
WordCounter.java
Console

```java
package com.wipro.komal.java8features;

import java.util.*;
//import java.util.stream.*;

public class WordCounter {
    public static void main(String[] args) {
        List<String> names = Arrays.asList("Komal", "Aniruddha", "Snehal", "Amit", "Bhavesh");

        // Count names longer than 5 characters
        long count = names.stream()
                          .filter(name -> name.length() > 5)
                          .count();

        System.out.println("Number of names longer than 5 characters: " + count);
    }
}
```

```
<terminated> WordCounter [Java Application] C:\sts-4.31.0.RELEASE\plugins\org.eclipse
Number of names longer than 5 characters: 3
```