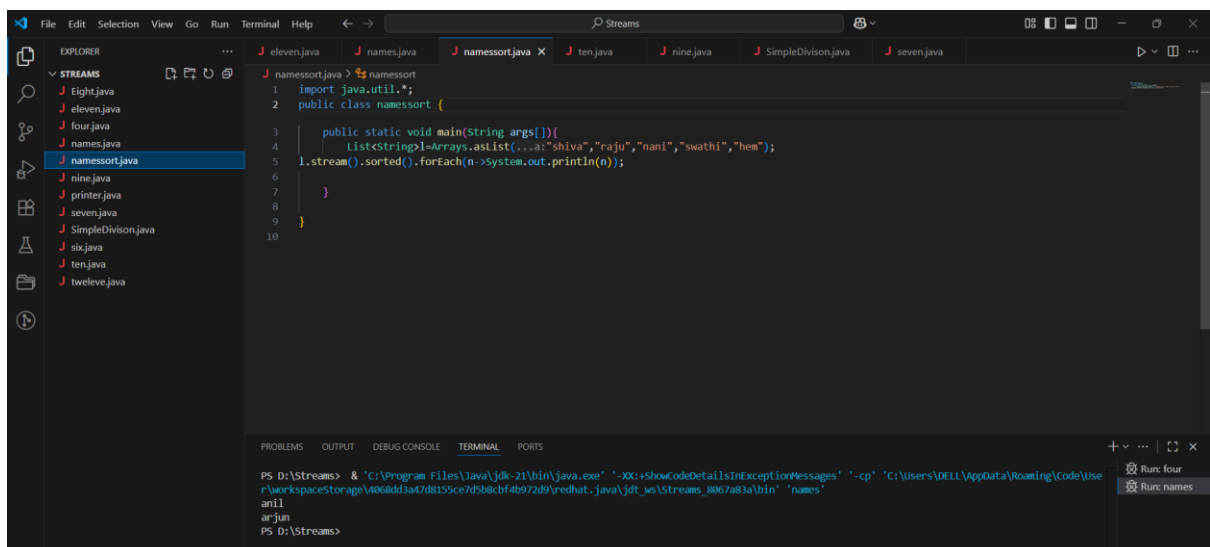
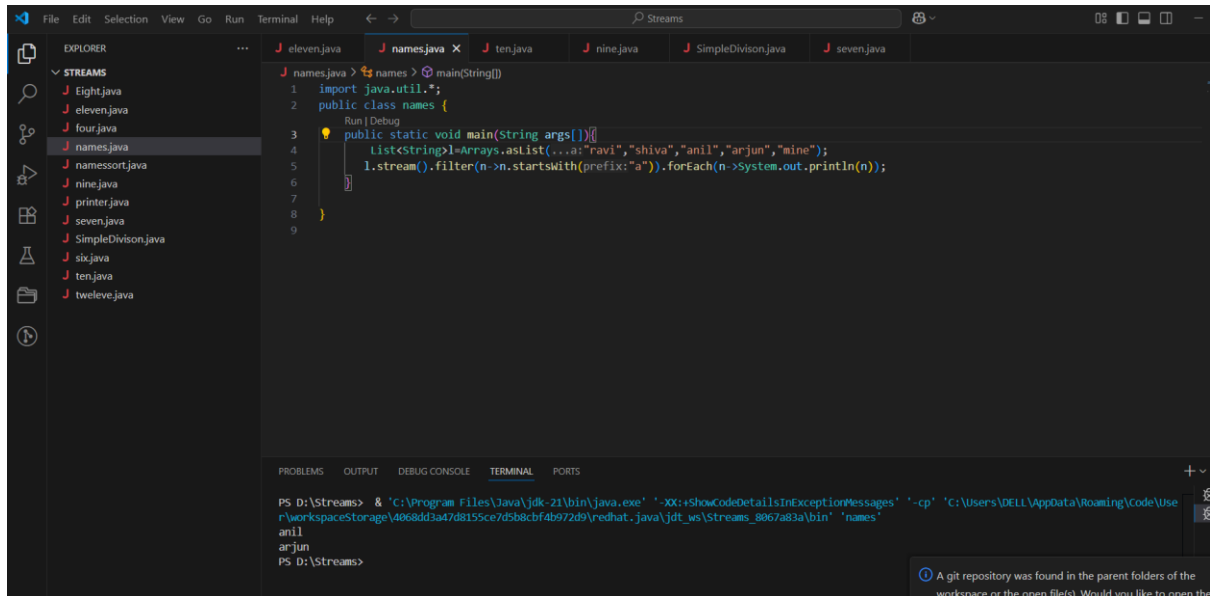
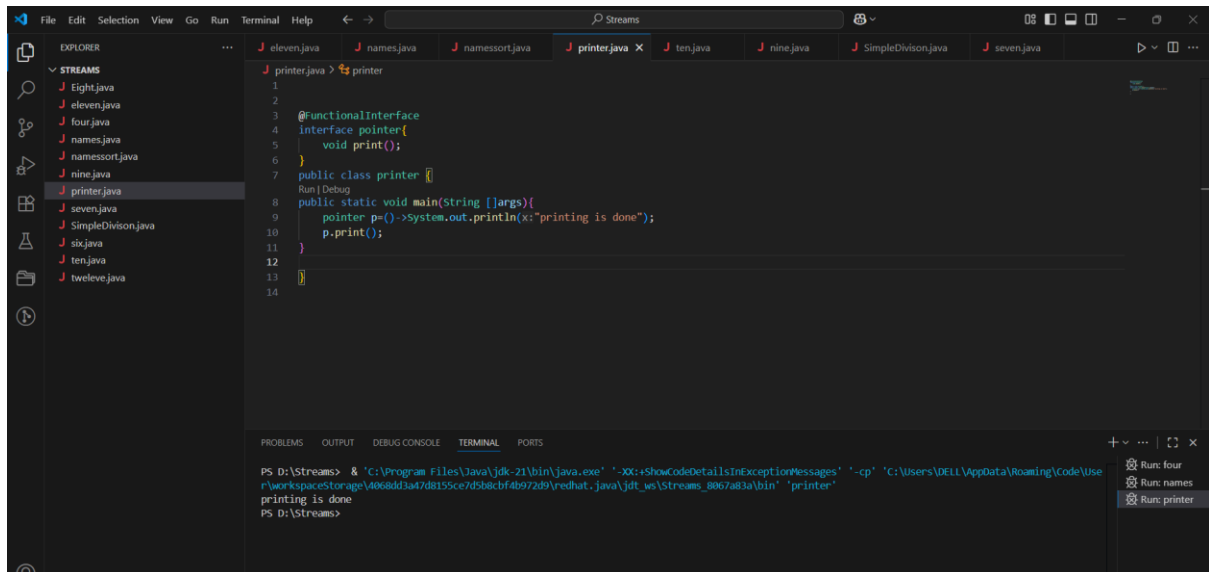


Streams Programs

Names that starts with a using filter



Functional Interface

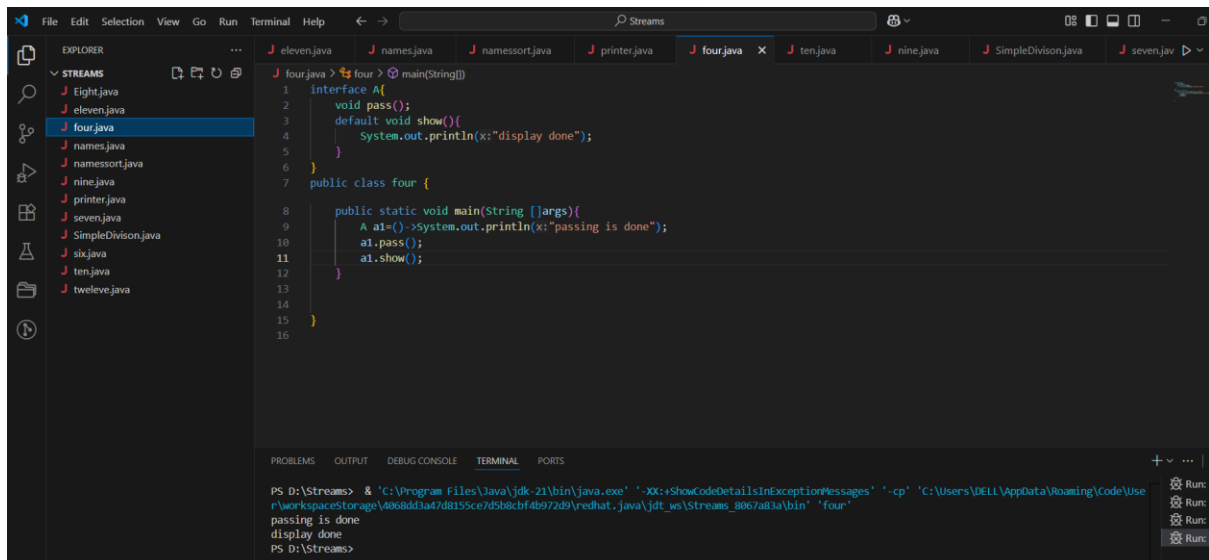


The screenshot shows an IDE with a project named 'STREAMS'. The 'EXPLORER' pane on the left lists several Java files, including 'printer.java'. The main editor displays the code for 'printer.java', which defines a functional interface 'pointer' and a class 'printer' that implements it. The 'pointer' interface has a single method 'print()'. The 'printer' class has a 'main' method that creates a 'pointer' object and calls 'print()', which outputs 'printing is done'.

```
1
2
3 @FunctionalInterface
4 interface pointer{
5     void print();
6 }
7 public class printer {
8     public static void main(String []args){
9         pointer p()->System.out.println(x:"printing is done");
10        p.print();
11    }
12 }
13
14
```

The 'TERMINAL' pane at the bottom shows the command used to run the program: `PS D:\Streams> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\DELL\AppData\Roaming\Code\Use r\workspaceStorage\4068dd3a47d8155ce7d5b8cf4b972d9\redhat.java\jdt_ws\Streams_8067a83a\bin' 'printer'`. The output shows 'printing is done'.

Default Methods

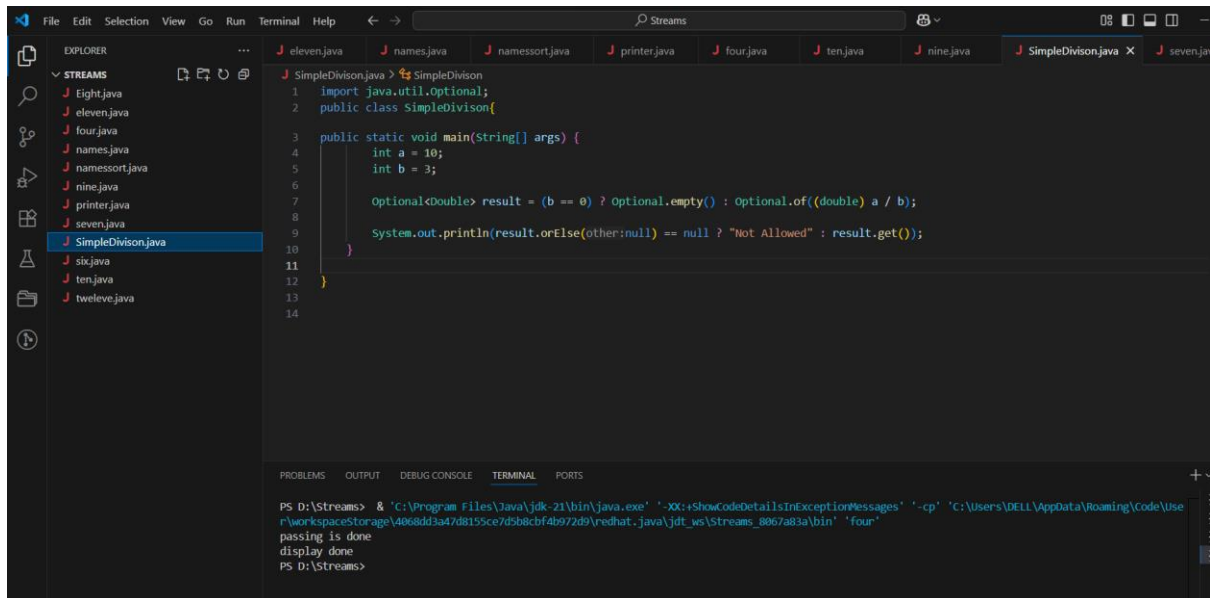


The screenshot shows an IDE with a project named 'STREAMS'. The 'EXPLORER' pane on the left lists several Java files, including 'four.java'. The main editor displays the code for 'four.java', which defines an interface 'A' with two methods: 'pass()' and 'show()'. The 'show()' method is a default method that outputs 'display done'. The 'four' class implements the 'A' interface and has a 'main' method that creates an instance of 'A', calls 'pass()', and then calls 'show()', which outputs 'display done'.

```
1 interface A{
2     void pass();
3     default void show(){
4         System.out.println(x:"display done");
5     }
6 }
7 public class four {
8     public static void main(String []args){
9         A a1()->System.out.println(x:"passing is done");
10        a1.pass();
11        a1.show();
12    }
13 }
14
15
16
```

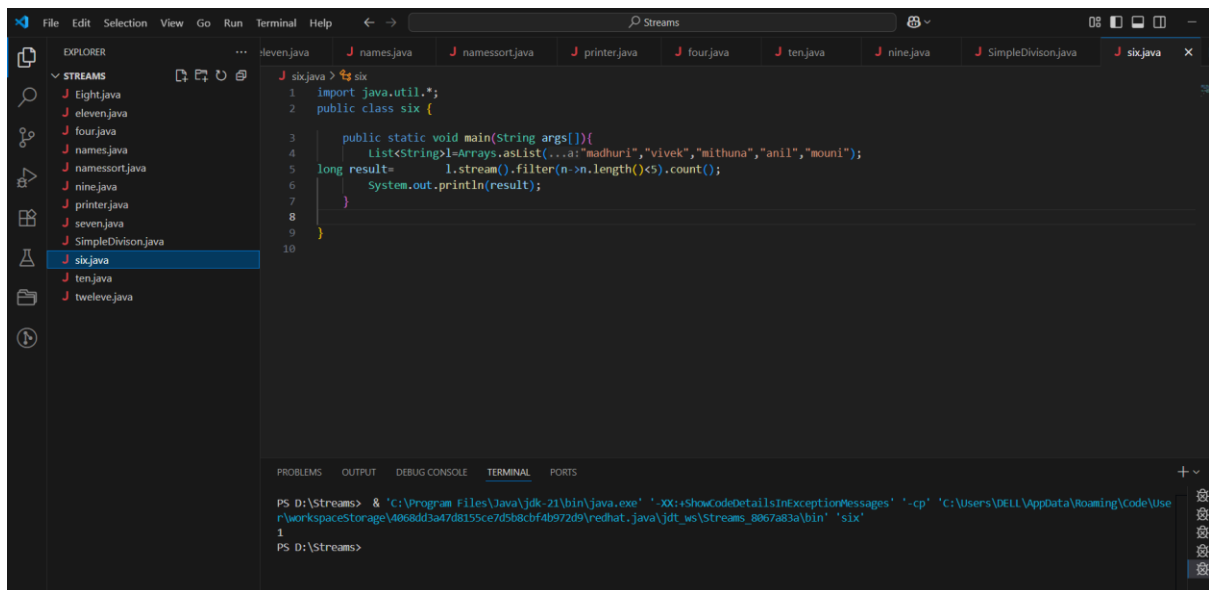
The 'TERMINAL' pane at the bottom shows the command used to run the program: `PS D:\Streams> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\DELL\AppData\Roaming\Code\Use r\workspaceStorage\4068dd3a47d8155ce7d5b8cf4b972d9\redhat.java\jdt_ws\Streams_8067a83a\bin' 'four'`. The output shows 'passing is done' and 'display done'.

Optional Class



```
File Edit Selection View Go Run Terminal Help  
SimpleDivison.java | eleven.java | names.java | namesort.java | printer.java | four.java | ten.java | nine.java | SimpleDivison.java | seven.java  
EXPLORER  
STREAMS  
Eight.java  
eleven.java  
four.java  
names.java  
namesort.java  
nine.java  
printer.java  
seven.java  
SimpleDivison.java  
six.java  
ten.java  
twelve.java  
SimpleDivison.java > SimpleDivison  
1 import java.util.Optional;  
2 public class SimpleDivison{  
3  
4     public static void main(String[] args) {  
5         int a = 10;  
6         int b = 3;  
7  
8         Optional<Double> result = (b == 0) ? Optional.empty() : Optional.of((double) a / b);  
9  
10        System.out.println(result.orElse(other:null) == null ? "Not Allowed" : result.get());  
11    }  
12  
13  
14  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
PS D:\Streams> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\DELL\AppData\Roaming\Code\Use  
r\workspaceStorage\4068dd3a47db155ce7d5b8cbf4b972d9\redhat_java\jdt_ws\Streams_8067a83a\bin' 'four'  
passing is done  
display done  
PS D:\Streams>
```

Filter The records where length less than 5



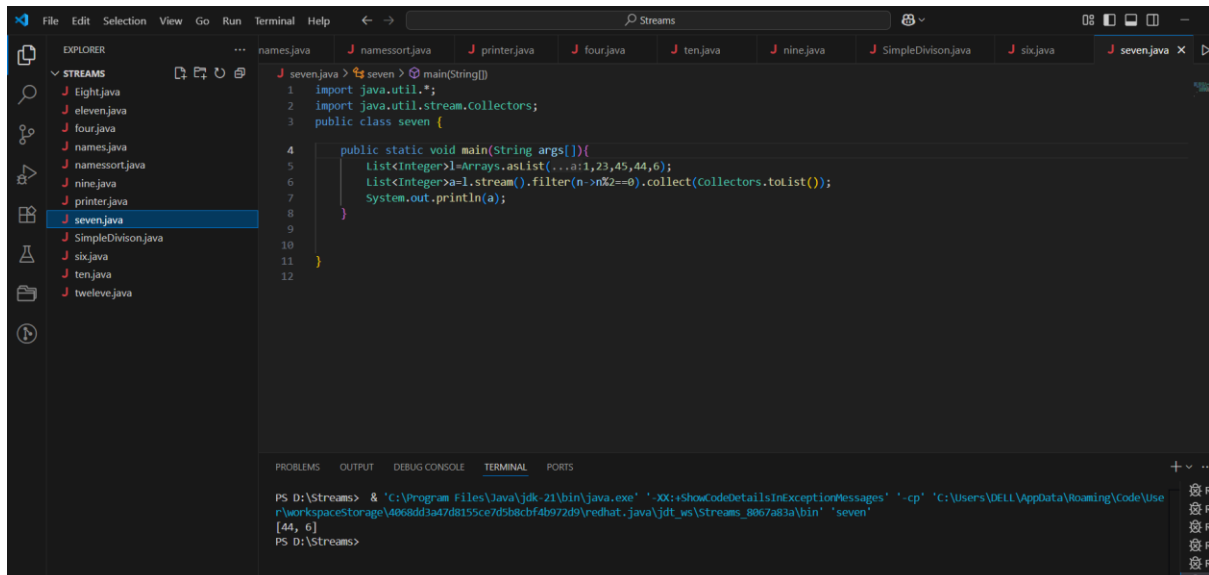
The screenshot shows an IDE with a project named 'Streams'. The Explorer panel on the left lists several Java files, including 'six.java'. The main editor displays the code for 'six.java':

```
1 import java.util.*;
2 public class six {
3
4     public static void main(String args[]){
5         List<String>l=Arrays.asList(...a:"madhuri","vivek","mithuna","anil","mouni");
6         long result=
7             l.stream().filter(n->n.length()<5).count();
8         System.out.println(result);
9     }
10 }
```

The bottom panel shows the terminal output:

```
PS D:\Streams> & 'C:\Program Files\Java\jdk-21\bin\java.exe' ^-XX:+ShowCodeDetailsInExceptionMessages' ^-cp 'C:\Users\DELL\AppData\Roaming\Code\Use
r\workspaceStorage\4068dd3a7d8155ce7d5b8cbf4b972d9\redhat_java\jdt_ws\Streams_8067a83a\bin' 'six'
1
PS D:\Streams>
```

Square root of integers using streams



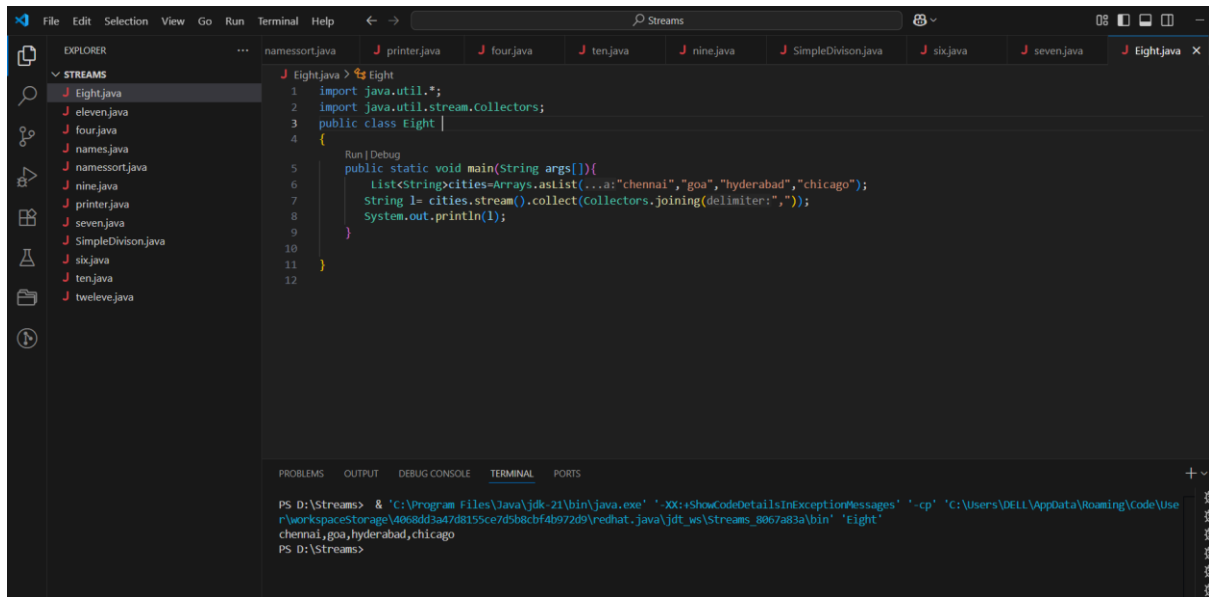
The screenshot shows the same IDE with a different Java file, 'seven.java', selected in the Explorer. The main editor displays the code for 'seven.java':

```
1 import java.util.*;
2 import java.util.stream.Collectors;
3 public class seven {
4
5     public static void main(String args[]){
6         List<Integer>l=Arrays.asList(...a:1,23,45,44,6);
7         List<Integer>a=l.stream().filter(n->n%2==0).collect(Collectors.toList());
8         System.out.println(a);
9     }
10 }
11
12 }
```

The bottom panel shows the terminal output:

```
PS D:\Streams> & 'C:\Program Files\Java\jdk-21\bin\java.exe' ^-XX:+ShowCodeDetailsInExceptionMessages' ^-cp 'C:\Users\DELL\AppData\Roaming\Code\Use
r\workspaceStorage\4068dd3a7d8155ce7d5b8cbf4b972d9\redhat_java\jdt_ws\Streams_8067a83a\bin' 'seven'
[44, 6]
PS D:\Streams>
```

Printing cities using collections



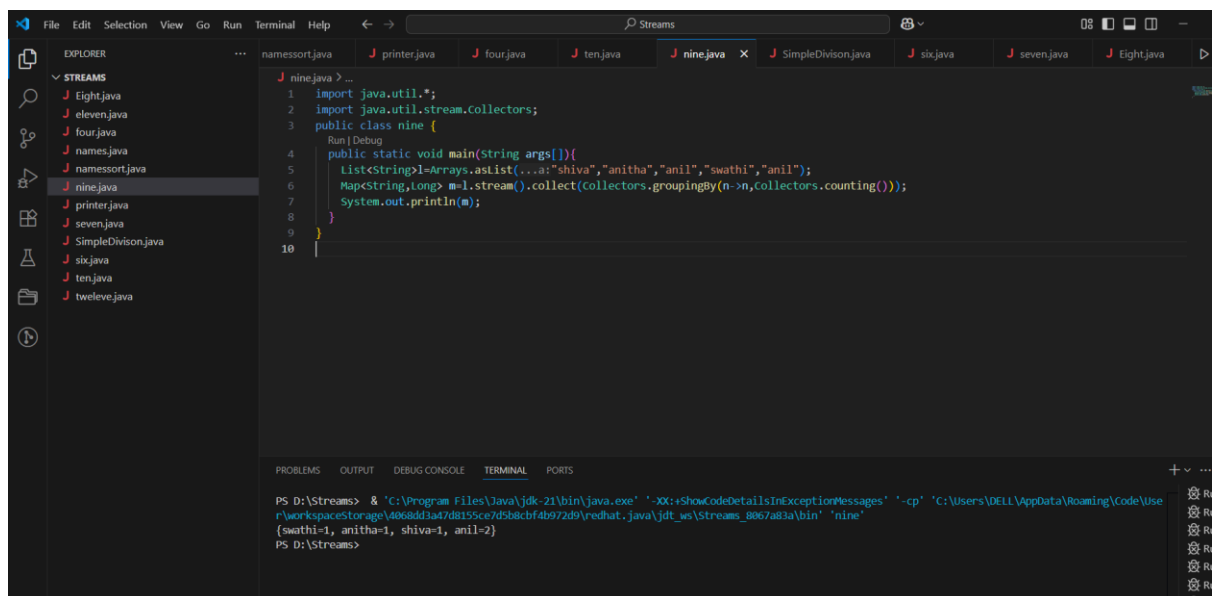
The screenshot shows an IDE with a project named 'Streams'. The 'EXPLORER' panel on the left lists several Java files, including 'Eight.java'. The 'EIGHT.java' file is open in the editor, showing the following code:

```
1 import java.util.*;
2 import java.util.stream.Collectors;
3 public class Eight {
4     {
5         Run | Debug
6         public static void main(String args[]){
7             List<String> cities=Arrays.asList(...:"chennai","goa","hyderabad","chicago");
8             String l= cities.stream().collect(Collectors.joining(delimiter:","));
9             System.out.println(l);
10        }
11    }
12 }
```

The 'TERMINAL' panel at the bottom shows the command used to run the program and its output:

```
PS D:\Streams> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\DELL\AppData\Roaming\Code\Use
r\workspaceStorage\4068dd3a47d8155ce7d5b8cbf4b972d9\redhat.java\jdt_ws\Streams_8067a83a\bin' 'Eight'
chennai,goa,hyderabad,chicago
PS D:\Streams>
```

Repeation of each records(Names)



The screenshot shows the same IDE with a different Java file, 'nine.java', open in the editor. The code is as follows:

```
1 import java.util.*;
2 import java.util.stream.Collectors;
3 public class nine {
4     {
5         Run | Debug
6         public static void main(String args[]){
7             List<String> l=Arrays.asList(...:"shiva","anitha","anil","swathi","anil");
8             Map<String,Long> m=l.stream().collect(Collectors.groupingBy(n->n,Collectors.counting()));
9             System.out.println(m);
10        }
11    }
12 }
```

The 'TERMINAL' panel shows the command and output for running this program:

```
PS D:\Streams> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\DELL\AppData\Roaming\Code\Use
r\workspaceStorage\4068dd3a47d8155ce7d5b8cbf4b972d9\redhat.java\jdt_ws\Streams_8067a83a\bin' 'nine'
{swathi=1, anitha=1, shiva=1, anil=2}
PS D:\Streams>
```

```
ten.java > ten > main(String[])
1 import java.util.*;
2 public class ten {
3     public static void main(String args[]) {
4         List<Integer> l = Arrays.asList(...a:1,45,34,22,12);
5         l.parallelStream().forEach(n->System.out.println(n*n));
6     }
7 }
8
9
```

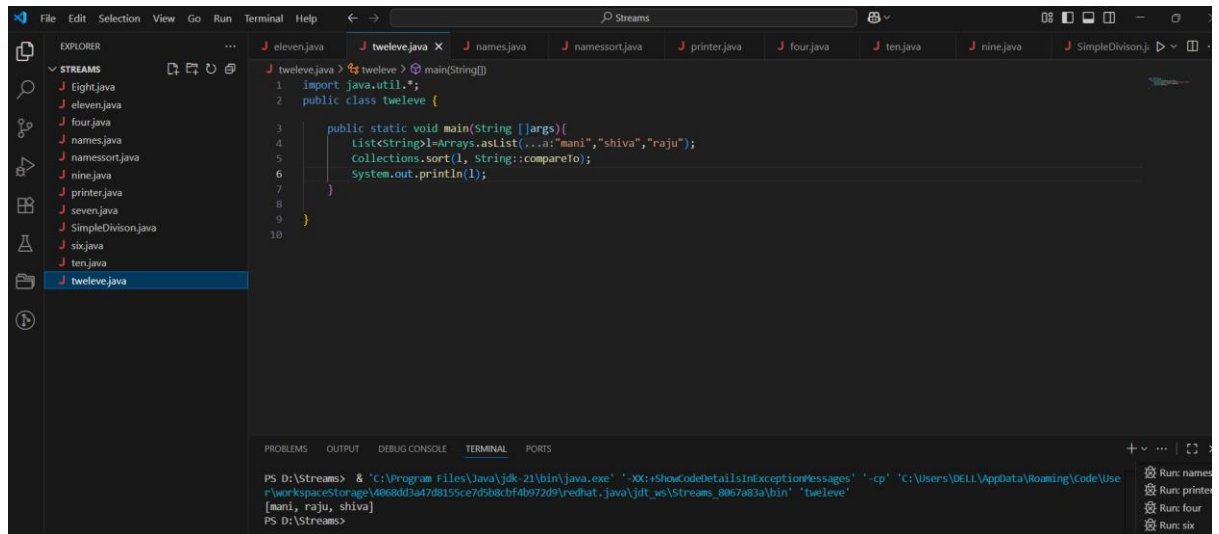
```
PS D:\Streams> & 'C:\Program Files\Java\jdk-21\bin\java.exe' ^XX:-ShowCodeDetailsInExceptionMessages' ^-cp 'C:\Users\DELL\AppData\Roaming\Code\Use
r\workspaceStorage\4068dd3a47d8155ce7d5b8cbf4b972d9\redhat.java\jdk_ws\Streams_8067a83a\bin' ^ten
1156
2025
484
144
1
PS D:\Streams>
```

Names that's starts and ending with same letter

```
eleven.java > eleven > main(String[])
1 import java.util.*;
2 import java.util.function.Predicate;
3 import java.util.stream.Collectors;
4 public class eleven {
5     public static void main(String args[]) {
6         List<String> l = Arrays.asList(...a:"anil","aruna","mamatha","swetha");
7         Predicate<String> p1=l1->l1.startsWith(prefix:"a");
8         Predicate<String> p2=l1->l1.endsWith(suffix:"a");
9         List<String> l2=l.stream().filter(p1.and(p2)).collect(Collectors.toList());
10        System.out.println(l2);
11    }
12 }
13
14
15
```

```
PS D:\Streams> & 'C:\Program Files\Java\jdk-21\bin\java.exe' ^XX:-ShowCodeDetailsInExceptionMessages' ^-cp 'C:\Users\DELL\AppData\Roaming\Code\Use
r\workspaceStorage\4068dd3a47d8155ce7d5b8cbf4b972d9\redhat.java\jdk_ws\Streams_8067a83a\bin' ^eleven
[aruna]
PS D:\Streams>
```

Sorting records based on alphabetic order



The screenshot shows an IDE with a project named 'Streams'. The Explorer panel on the left lists several Java files, with 'tweleve.java' selected. The main editor displays the code for 'tweleve.java', which imports 'java.util.*' and defines a 'main' method. Inside the 'main' method, an array of names is created using 'Arrays.asList(...)', and then sorted using 'Collections.sort()' with a comparator. The sorted array is printed using 'System.out.println()'. The bottom panel shows the 'TERMINAL' output, which displays the command used to run the program and the resulting output: '[mani, raj, shiva]'.

```
1 import java.util.*;
2 public class tweleve {
3
4     public static void main(String []args){
5         List<String>l=Arrays.asList(...a:"mani","shiva","raju");
6         Collections.sort(l, String::compareTo);
7         System.out.println(l);
8     }
9 }
10
```

PS D:\Streams> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-Xt:showCodeDetailsInExceptionMessages' '-cp' 'C:\Users\DELL\AppData\Roaming\Code\Use
r\workspaceStorage\486add3a4708155ce7d508cf4b972d9\redhat.java\jdt_ws\Streams_8067a83a\bin' "tweleve"
[mani, raj, shiva]
PS D:\Streams>