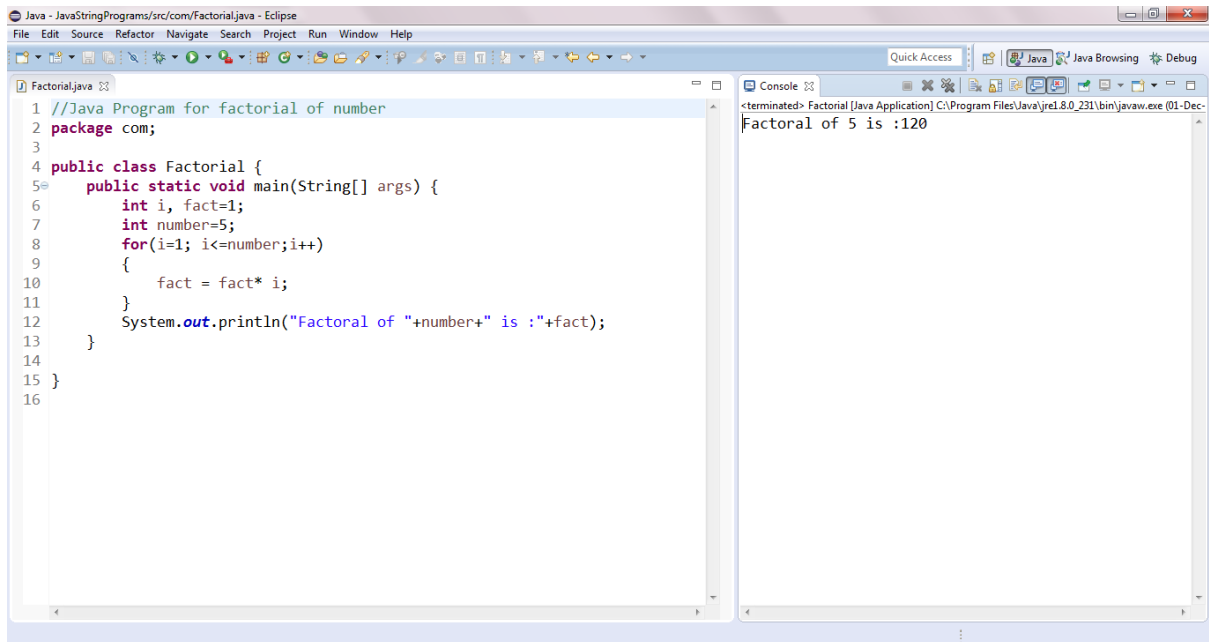


1) Factorial No

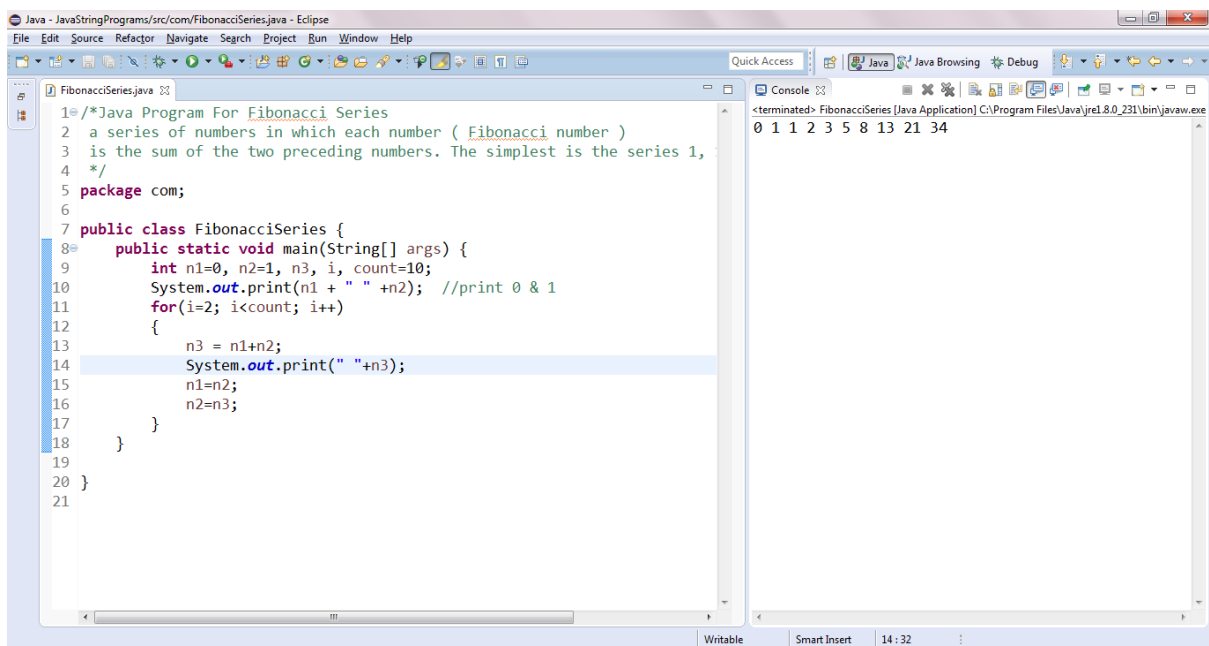


The screenshot shows the Eclipse IDE with a Java project named 'JavaStringPrograms'. The file 'Factorial.java' is open in the editor. The code calculates the factorial of 5. The console on the right shows the output: 'Factorial of 5 is :120'.

```
1 //Java Program for factorial of number
2 package com;
3
4 public class Factorial {
5     public static void main(String[] args) {
6         int i, fact=1;
7         int number=5;
8         for(i=1; i<=number; i++)
9         {
10             fact = fact* i;
11         }
12         System.out.println("Factorial of "+number+" is :"+fact);
13     }
14 }
15 }
16 }
```

Console Output:
<terminated> Factorial [Java Application] C:\Program Files\Java\jre1.8.0_231\bin\javaw.exe (01-Dec-2023 14:32)
Factorial of 5 is :120

2) Fibonacci Series

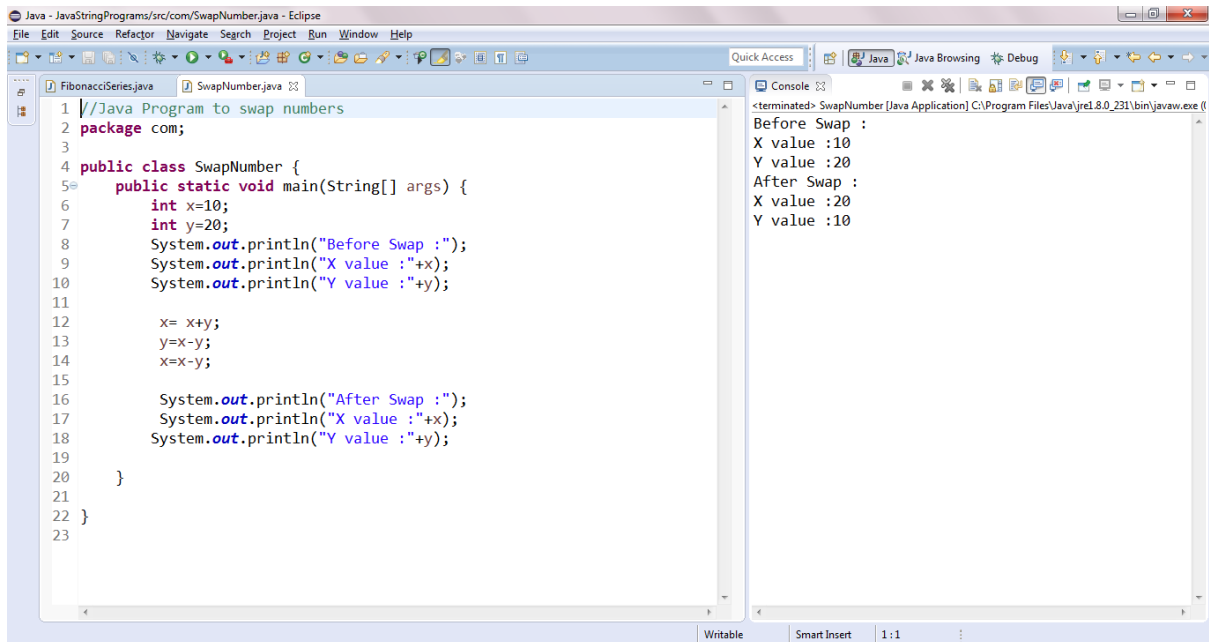


The screenshot shows the Eclipse IDE with a Java project named 'JavaStringPrograms'. The file 'FibonacciSeries.java' is open in the editor. The code generates the first 10 numbers of the Fibonacci series. The console on the right shows the output: '0 1 1 2 3 5 8 13 21 34'.

```
1 /*Java Program For Fibonacci Series
2 a series of numbers in which each number ( Fibonacci number )
3 is the sum of the two preceding numbers. The simplest is the series 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...
4 */
5 package com;
6
7 public class FibonacciSeries {
8     public static void main(String[] args) {
9         int n1=0, n2=1, n3, i, count=10;
10        System.out.print(n1 + " " + n2); //print 0 & 1
11        for(i=2; i<count; i++)
12        {
13            n3 = n1+n2;
14            System.out.print(" " + n3);
15            n1=n2;
16            n2=n3;
17        }
18    }
19 }
20 }
21 }
```

Console Output:
<terminated> FibonacciSeries [Java Application] C:\Program Files\Java\jre1.8.0_231\bin\javaw.exe (01-Dec-2023 14:32)
0 1 1 2 3 5 8 13 21 34

3) Swap Numbers

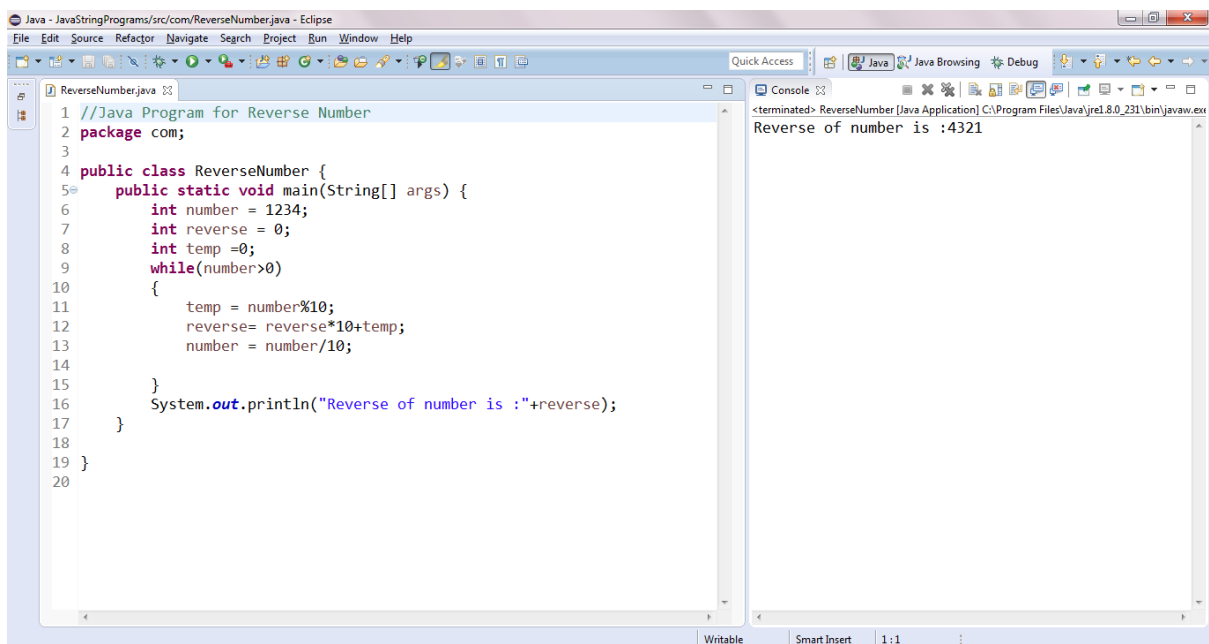


```
1 //Java Program to swap numbers
2 package com;
3
4 public class SwapNumber {
5     public static void main(String[] args) {
6         int x=10;
7         int y=20;
8         System.out.println("Before Swap :");
9         System.out.println("X value :"+x);
10        System.out.println("Y value :"+y);
11
12        x= x+y;
13        y=x-y;
14        x=x-y;
15
16        System.out.println("After Swap :");
17        System.out.println("X value :"+x);
18        System.out.println("Y value :"+y);
19    }
20 }
21
22 }
23
```

Console Output:

```
<terminated> SwapNumber [Java Application] C:\Program Files\Java\jre1.8.0_231\bin\javaw.exe (
Before Swap :
X value :10
Y value :20
After Swap :
X value :20
Y value :10
```

4) Reverse No

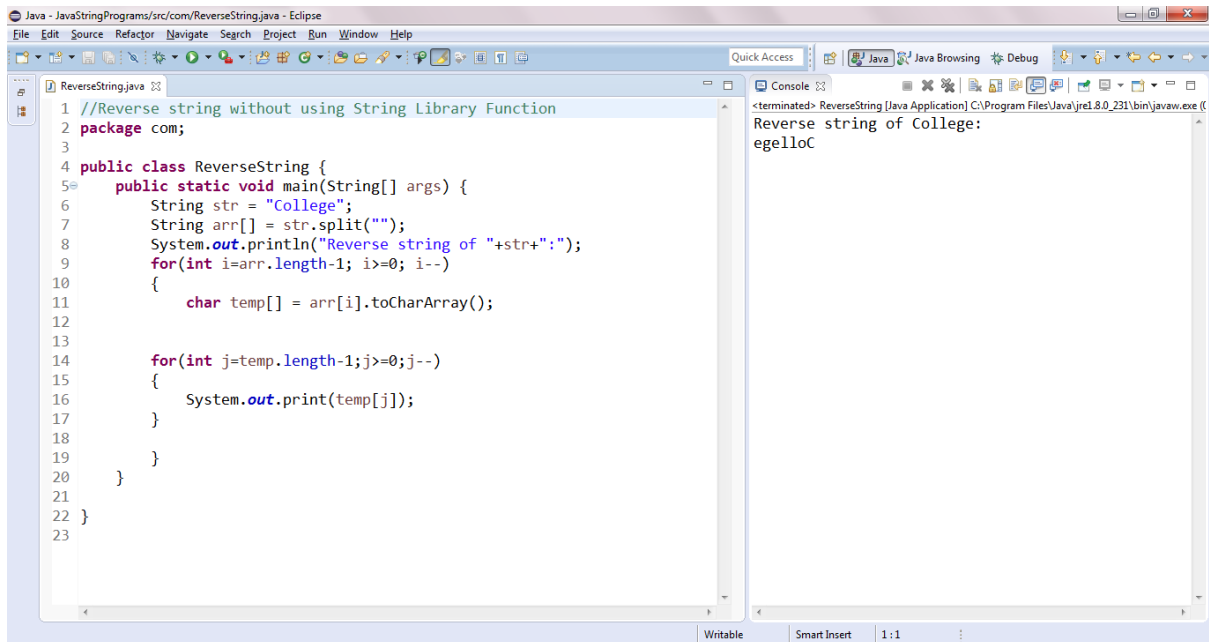


```
1 //Java Program for Reverse Number
2 package com;
3
4 public class ReverseNumber {
5     public static void main(String[] args) {
6         int number = 1234;
7         int reverse = 0;
8         int temp =0;
9         while(number>0)
10        {
11            temp = number%10;
12            reverse= reverse*10+temp;
13            number = number/10;
14        }
15        System.out.println("Reverse of number is :"+reverse);
16    }
17 }
18
19 }
20
```

Console Output:

```
<terminated> ReverseNumber [Java Application] C:\Program Files\Java\jre1.8.0_231\bin\javaw.exe
Reverse of number is :4321
```

5) Reverse String



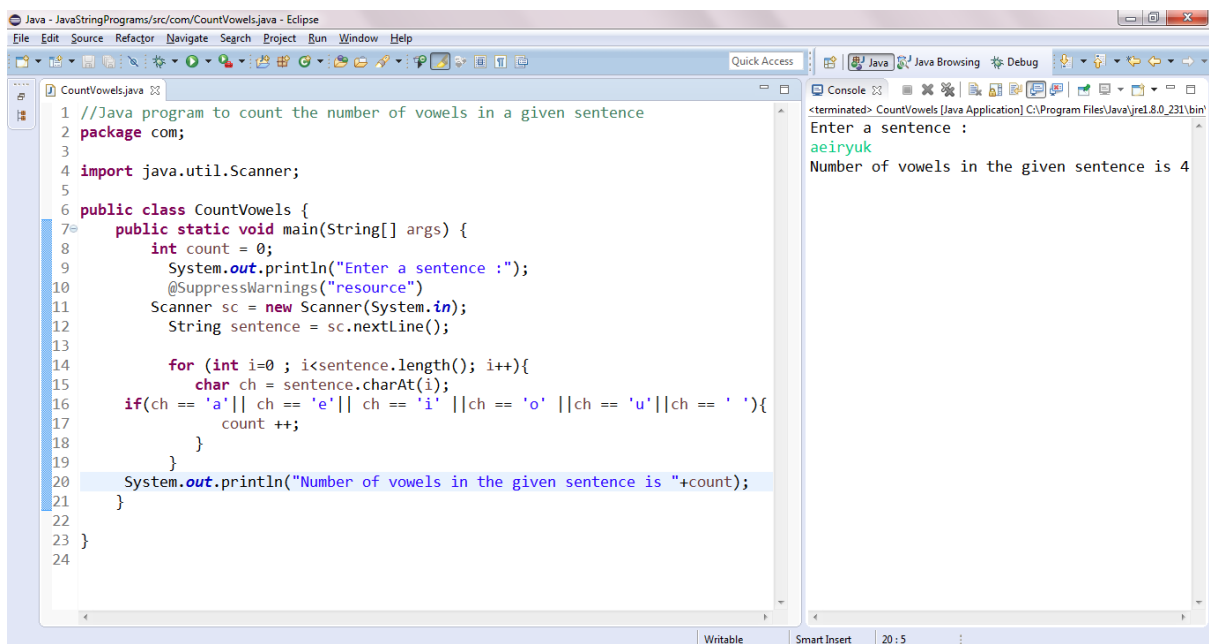
The screenshot shows the Eclipse IDE with the file `ReverseString.java` open. The code is as follows:

```
1 //Reverse string without using String Library Function
2 package com;
3
4 public class ReverseString {
5     public static void main(String[] args) {
6         String str = "College";
7         String arr[] = str.split("");
8         System.out.println("Reverse string of "+str+":");
9         for(int i=arr.length-1; i>=0; i--){
10             {
11                 char temp[] = arr[i].toCharArray();
12
13                 for(int j=temp.length-1;j>=0;j--){
14                     {
15                         System.out.print(temp[j]);
16                     }
17                 }
18             }
19         }
20     }
21 }
22 }
23 }
```

The console output on the right shows the program's execution:

```
<terminated> ReverseString [Java Application] C:\Program Files\Java\jre1.8.0_231\bin\javaw.exe (
Reverse string of College:
egelloC
```

6) Count Vowels in String



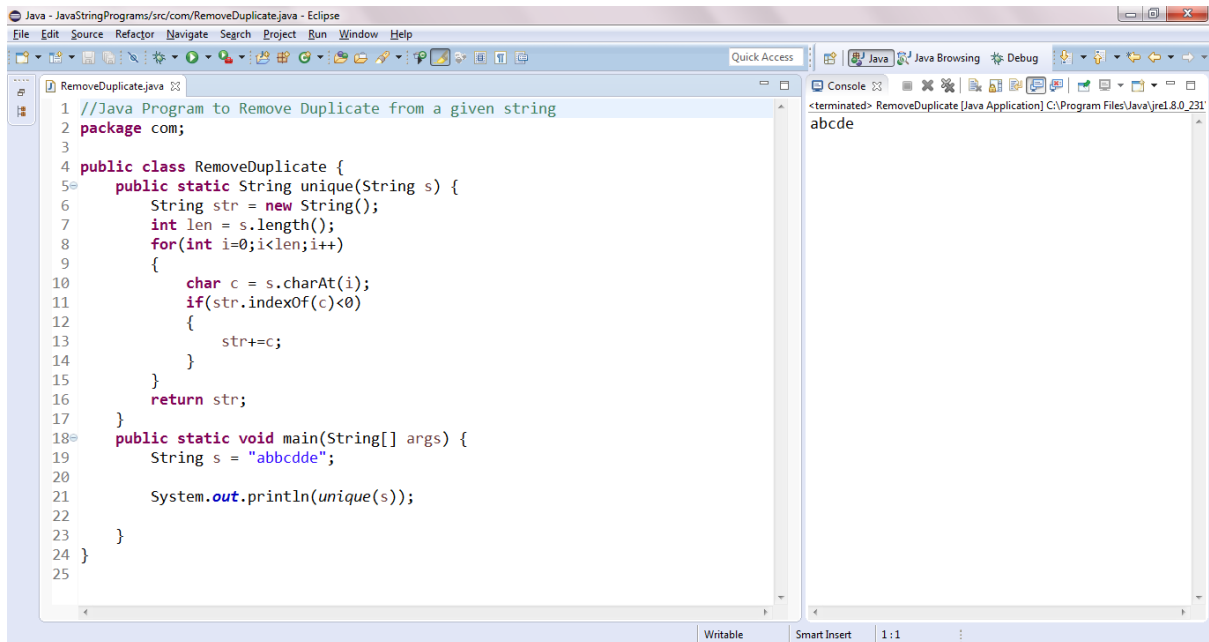
The screenshot shows the Eclipse IDE with the file `CountVowels.java` open. The code is as follows:

```
1 //Java program to count the number of vowels in a given sentence
2 package com;
3
4 import java.util.Scanner;
5
6 public class CountVowels {
7     public static void main(String[] args) {
8         int count = 0;
9         System.out.println("Enter a sentence :");
10        @SuppressWarnings("resource")
11        Scanner sc = new Scanner(System.in);
12        String sentence = sc.nextLine();
13
14        for (int i=0 ; i<sentence.length(); i++){
15            char ch = sentence.charAt(i);
16            if(ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' || ch == ' '){
17                count ++;
18            }
19        }
20        System.out.println("Number of vowels in the given sentence is "+count);
21    }
22 }
23 }
24 }
```

The console output on the right shows the program's execution:

```
<terminated> CountVowels [Java Application] C:\Program Files\Java\jre1.8.0_231\bin\
Enter a sentence :
aeiryuk
Number of vowels in the given sentence is 4
```

7) Remove Duplicates character in String

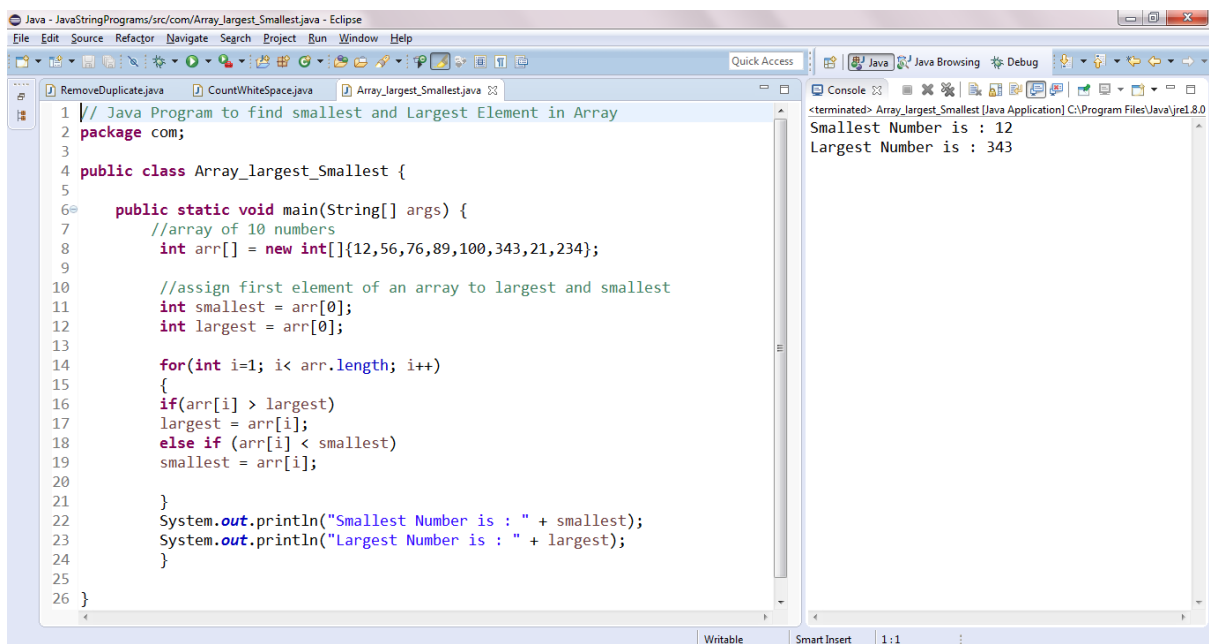


The screenshot shows the Eclipse IDE with a Java project named 'JavaStringPrograms'. The file 'RemoveDuplicate.java' is open in the editor. The code defines a class 'RemoveDuplicate' with a static method 'unique' that takes a string 's' and returns a string with all duplicate characters removed. The 'main' method calls 'unique' on the string 'abdcde' and prints the result. The console on the right shows the output 'abcde'.

```
1 //Java Program to Remove Duplicate from a given string
2 package com;
3
4 public class RemoveDuplicate {
5     public static String unique(String s) {
6         String str = new String();
7         int len = s.length();
8         for(int i=0;i<len;i++)
9         {
10             char c = s.charAt(i);
11             if(str.indexOf(c)<0)
12             {
13                 str+=c;
14             }
15         }
16         return str;
17     }
18     public static void main(String[] args) {
19         String s = "abdcde";
20
21         System.out.println(unique(s));
22     }
23 }
24
25
```

Console Output: abcde

8) Java Program to find smallest and Largest Elements in Array

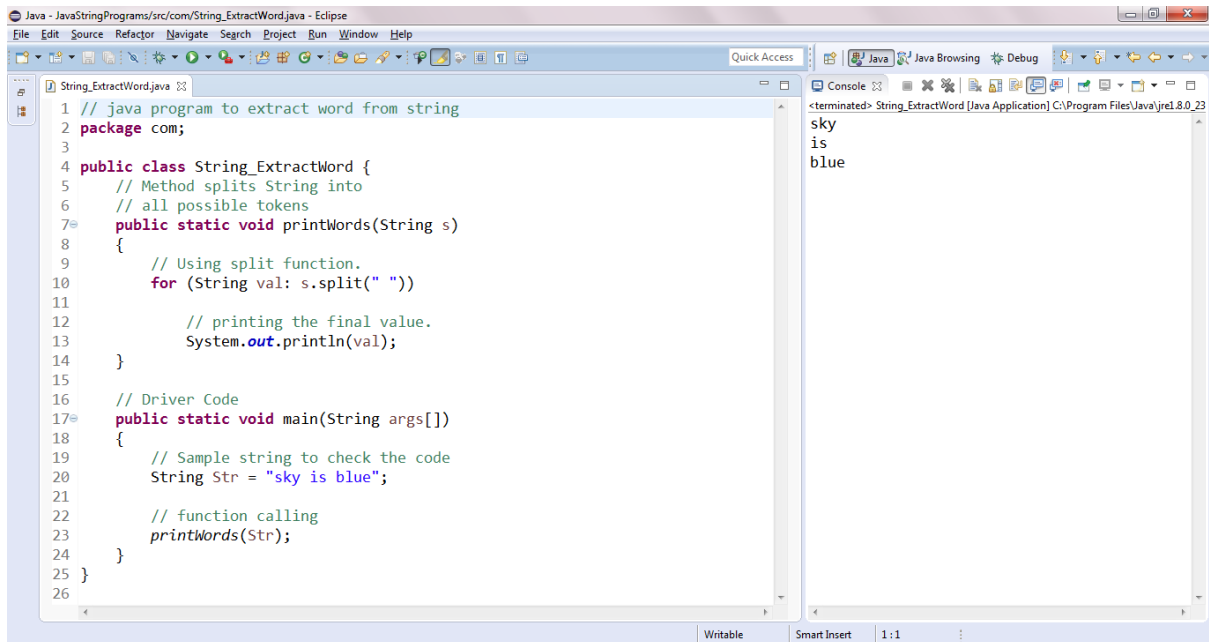


The screenshot shows the Eclipse IDE with a Java project named 'JavaStringPrograms'. The file 'Array_largest_Smallest.java' is open in the editor. The code defines a class 'Array_largest_Smallest' with a static method 'main' that takes an array of integers and finds the smallest and largest elements. The 'main' method initializes an array of 10 numbers and prints the smallest and largest values. The console on the right shows the output 'Smallest Number is : 12' and 'Largest Number is : 343'.

```
1 // Java Program to find smallest and Largest Element in Array
2 package com;
3
4 public class Array_largest_Smallest {
5
6     public static void main(String[] args) {
7         //array of 10 numbers
8         int arr[] = new int[]{12,56,76,89,100,343,21,234};
9
10        //assign first element of an array to largest and smallest
11        int smallest = arr[0];
12        int largest = arr[0];
13
14        for(int i=1; i< arr.length; i++)
15        {
16            if(arr[i] > largest)
17                largest = arr[i];
18            else if (arr[i] < smallest)
19                smallest = arr[i];
20        }
21        System.out.println("Smallest Number is : " + smallest);
22        System.out.println("Largest Number is : " + largest);
23    }
24 }
25
26
```

Console Output: Smallest Number is : 12
Largest Number is : 343

9) Java Program to Extract word from string



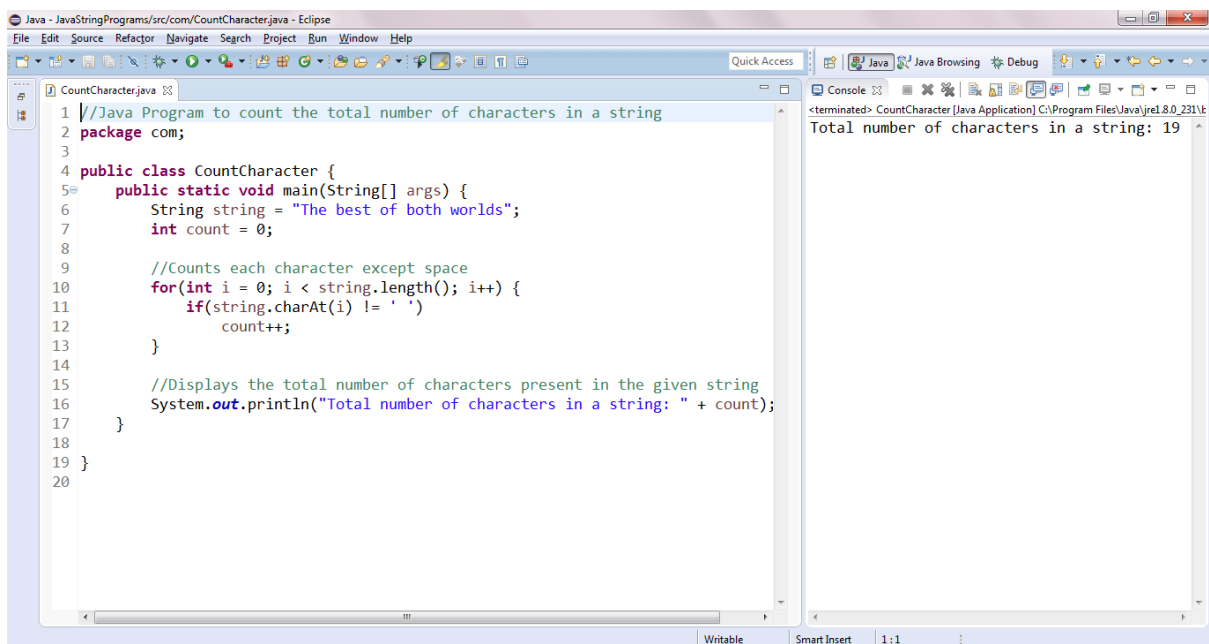
The screenshot shows the Eclipse IDE with a Java project named 'JavaStringPrograms'. The editor displays the file 'String_ExtractWord.java'. The code defines a class 'String_ExtractWord' with a static method 'printWords' that splits a string by spaces and prints each token. The 'main' method uses a sample string 'sky is blue' and calls 'printWords'. The console on the right shows the output: 'sky', 'is', and 'blue' on separate lines.

```
1 // java program to extract word from string
2 package com;
3
4 public class String_ExtractWord {
5     // Method splits String into
6     // all possible tokens
7     public static void printWords(String s)
8     {
9         // Using split function.
10        for (String val: s.split(" "))
11        {
12            // printing the final value.
13            System.out.println(val);
14        }
15    }
16    // Driver Code
17    public static void main(String args[])
18    {
19        // Sample string to check the code
20        String Str = "sky is blue";
21
22        // function calling
23        printWords(Str);
24    }
25 }
26
```

Console Output:

```
<terminated> String_ExtractWord [Java Application] C:\Program Files\Java\jre1.8.0_231\bin\java.exe
sky
is
blue
```

10) Java Program to count the total number of character in a string



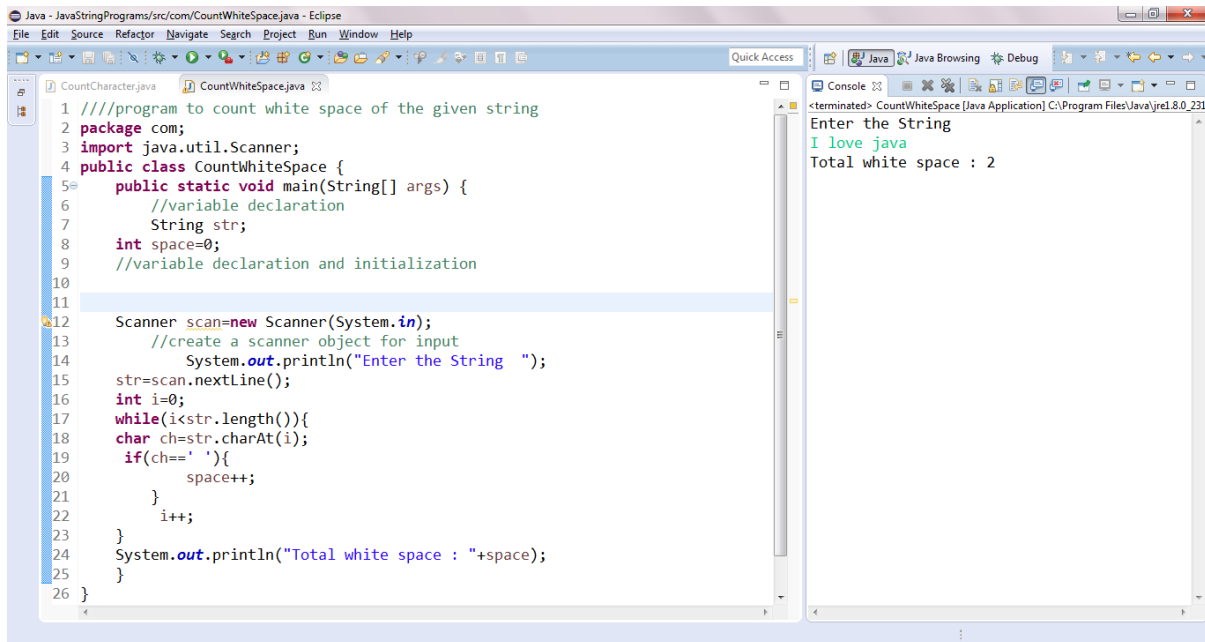
The screenshot shows the Eclipse IDE with a Java project named 'JavaStringPrograms'. The editor displays the file 'CountCharacter.java'. The code defines a class 'CountCharacter' with a static method 'main' that counts the number of non-space characters in the string 'The best of both worlds'. The console on the right shows the output: 'Total number of characters in a string: 19'.

```
1 //Java Program to count the total number of characters in a string
2 package com;
3
4 public class CountCharacter {
5     public static void main(String[] args) {
6         String string = "The best of both worlds";
7         int count = 0;
8
9         //Counts each character except space
10        for(int i = 0; i < string.length(); i++) {
11            if(string.charAt(i) != ' ')
12                count++;
13        }
14
15        //Displays the total number of characters present in the given string
16        System.out.println("Total number of characters in a string: " + count);
17    }
18 }
19 }
20
```

Console Output:

```
<terminated> CountCharacter [Java Application] C:\Program Files\Java\jre1.8.0_231\bin\java.exe
Total number of characters in a string: 19
```

11) Java Program to count white space of the given string



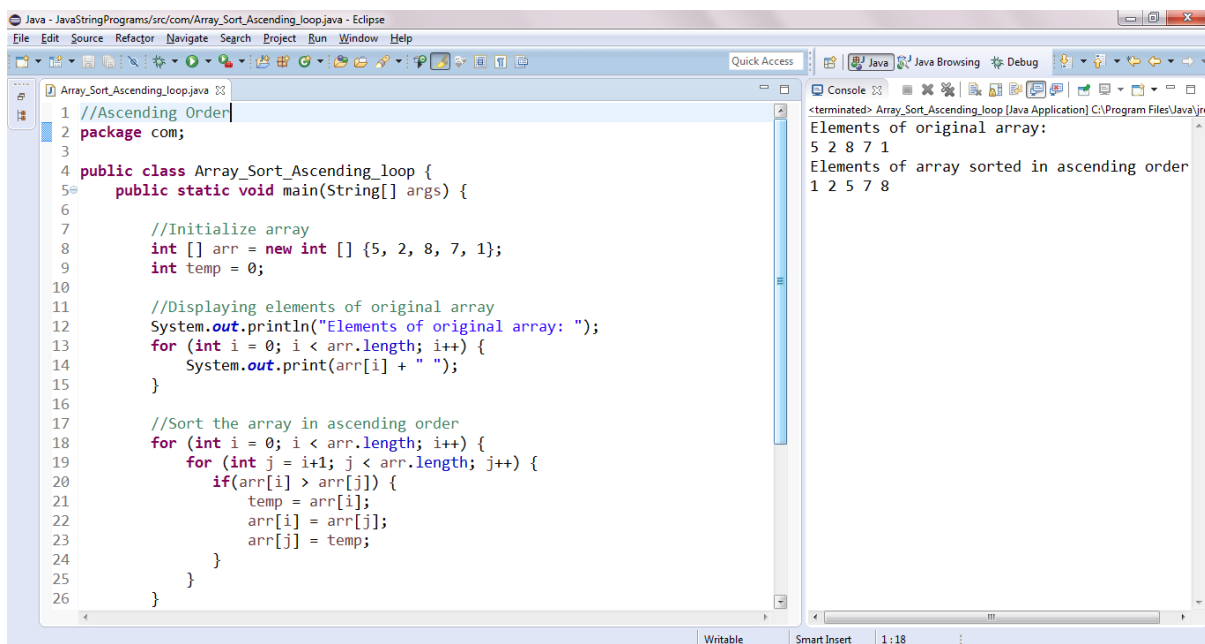
The screenshot shows the Eclipse IDE with a Java project named 'JavaStringPrograms'. The file 'CountWhiteSpace.java' is open in the editor. The code defines a class 'CountWhiteSpace' with a 'main' method that uses a 'Scanner' to read a string from the user and counts the number of white spaces. The console output shows the user entered 'I love java' and the program outputted 'Total white space : 2'.

```
1  ///program to count white space of the given string
2  package com;
3  import java.util.Scanner;
4  public class CountWhiteSpace {
5      public static void main(String[] args) {
6          //variable declaration
7          String str;
8          int space=0;
9          //variable declaration and initialization
10
11
12      Scanner scan=new Scanner(System.in);
13          //create a scanner object for input
14          System.out.println("Enter the String ");
15      str=scan.nextLine();
16      int i=0;
17      while(i<str.length()){
18          char ch=str.charAt(i);
19          if(ch==' '){
20              space++;
21          }
22          i++;
23      }
24      System.out.println("Total white space : "+space);
25  }
26 }
```

Console Output:

```
<terminated> CountWhiteSpace [Java Application] C:\Program Files\Java\jre1.8.0_231
Enter the String
I love java
Total white space : 2
```

12) Sort the Array in Ascending Order



The screenshot shows the Eclipse IDE with a Java project named 'JavaStringPrograms'. The file 'Array_Sort_Ascending_loop.java' is open in the editor. The code defines a class 'Array_Sort_Ascending_loop' with a 'main' method that initializes an array, displays its elements, and sorts it in ascending order using a nested loop. The console output shows the original array '5 2 8 7 1' and the sorted array '1 2 5 7 8'.

```
1  //Ascending Order
2  package com;
3
4  public class Array_Sort_Ascending_loop {
5      public static void main(String[] args) {
6
7          //Initialize array
8          int [] arr = new int [] {5, 2, 8, 7, 1};
9          int temp = 0;
10
11          //Displaying elements of original array
12          System.out.println("Elements of original array: ");
13          for (int i = 0; i < arr.length; i++) {
14              System.out.print(arr[i] + " ");
15          }
16
17          //Sort the array in ascending order
18          for (int i = 0; i < arr.length; i++) {
19              for (int j = i+1; j < arr.length; j++) {
20                  if(arr[i] > arr[j]) {
21                      temp = arr[i];
22                      arr[i] = arr[j];
23                      arr[j] = temp;
24                  }
25              }
26          }
27      }
28  }
```

Console Output:

```
<terminated> Array_Sort_Ascending_loop [Java Application] C:\Program Files\Java\jre1.8.0_231
Elements of original array:
5 2 8 7 1
Elements of array sorted in ascending order
1 2 5 7 8
```

```
13     for (int i = 0; i < arr.length; i++) {
14         System.out.print(arr[i] + " ");
15     }
16
17     //Sort the array in ascending order
18     for (int i = 0; i < arr.length; i++) {
19         for (int j = i+1; j < arr.length; j++) {
20             if(arr[i] > arr[j]) {
21                 temp = arr[i];
22                 arr[i] = arr[j];
23                 arr[j] = temp;
24             }
25         }
26     }
27
28     System.out.println();
29
30     //Displaying elements of array after sorting
31     System.out.println("Elements of array sorted in ascending order: ");
32     for (int i = 0; i < arr.length; i++) {
33         System.out.print(arr[i] + " ");
34     }
35 }
36
37 }
38
```

Console Output:

```
<terminated> Array_Sort_Ascending_loop [Java Application] C:\Program Files\Java\jre1.8.0_...
Elements of original array:
5 2 8 7 1
Elements of array sorted in ascending order
1 2 5 7 8
```

Sort the Array in Decending Order

```
1 //Java Program to sort array element in decending order
2 package com;
3
4 public class Array_Sor_Descending {
5     public static void main(String[] args) {
6
7
8         //Initialize array
9         int [] arr = new int [] {5, 2, 8, 7, 1};
10        int temp = 0;
11
12        //Displaying elements of original array
13        System.out.println("Elements of original array: ");
14        for (int i = 0; i < arr.length; i++) {
15            System.out.print(arr[i] + " ");
16        }
17
18        //Sort the array in descending order
19        for (int i = 0; i < arr.length; i++) {
20            for (int j = i+1; j < arr.length; j++) {
21                if(arr[i] < arr[j]) {
22                    temp = arr[i];
23                    arr[i] = arr[j];
24                    arr[j] = temp;
25                }
26            }
27        }
28    }
29 }
30
```

Console Output:

```
<terminated> Array_Sor_Descending [Java Application] C:\Program Files\Java\jre1.8.0_...
Elements of original array:
5 2 8 7 1
Elements of array sorted in descending order
8 7 5 2 1
```

Java - JavaStringPrograms/src/com/Array_Sor_Descending.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access

Array_Sor_Descending.java

```
13 System.out.println("Elements of original array: ");
14 for (int i = 0; i < arr.length; i++) {
15     System.out.print(arr[i] + " ");
16 }
17
18 //Sort the array in descending order
19 for (int i = 0; i < arr.length; i++) {
20     for (int j = i+1; j < arr.length; j++) {
21         if(arr[i] < arr[j]) {
22             temp = arr[i];
23             arr[i] = arr[j];
24             arr[j] = temp;
25         }
26     }
27 }
28
29 System.out.println();
30
31 //Displaying elements of array after sorting
32 System.out.println("Elements of array sorted in descending order: ");
33 for (int i = 0; i < arr.length; i++) {
34     System.out.print(arr[i] + " ");
35 }
36 }
37 }
38 }
```

Console

<terminated> Array_Sor_Descending [Java Application] C:\Program Files\Java\jre1.8.0_231\bin\java.exe

Elements of original array:
5 2 8 7 1
Elements of array sorted in descending order:
8 7 5 2 1

Writable Smart Insert 1:1

Java - JavaStringPrograms/src/com/Array_Sorting.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access

Array_Sor_Descending.java Array_Sorting.java

```
1 //Sort Numeric Array In Ascending Order
2 package com;
3
4 import java.util.Arrays;
5
6 public class Array_Sorting {
7
8     public static void main(String[] args)
9     {
10         //define an array
11         int[] intArray = {52, 45, 32, 64, 12, 87, 78, 98, 23, 7};
12         System.out.printf("Original Array : %s", Arrays.toString(intArray));
13
14         Arrays.sort(intArray);
15         System.out.printf("\n\nSorted Array : %s", Arrays.toString(intArray));
16     }
17 }
18
19
20
```

Console

<terminated> Array_Sorting [Java Application] C:\Program Files\Java\jre1.8.0_231\bin\java.exe

Original Array : [52, 45, 32, 64, 12, 87, 78, 98, 23, 7]
Sorted Array : [7, 12, 23, 32, 45, 52, 64, 78, 87, 98]

Writable Smart Insert 15:8

The screenshot shows the Eclipse IDE with a Java project named 'JavaStringPrograms'. The editor displays the file 'ArraySortDecendingOrder.java' (note the typo in the filename). The code defines a class 'ArraySortDecendingOrder' with a 'main' method. It imports 'java.util.Arrays' and defines an integer array 'IntArray' with values {52, 45, 32, 64, 12, 87, 78, 98, 23, 7}. The program prints the original array, sorts it in descending order using 'Arrays.sort(IntArray, Collections.reverseOrder())', and prints the sorted array. The console output shows the original array and the sorted array: [98, 87, 78, 64, 52, 45, 32, 23, 12, 7].

```
1 //Decending Order
2 package com;
3
4 import java.util.Arrays;
5
6
7 public class ArraySortDecendingOrder {
8     public static void main(String[] args)
9     {
10         //Collections.reverseOrder do not work for primitive Types
11         //define an array with Integer
12         Integer[] IntArray = {52, 45, 32, 64, 12, 87, 78, 98, 23, 7};
13
14         //print original array
15         System.out.printf("Original Array: %s",
16             Arrays.toString(IntArray));
17
18         // Sorts IntArray in descending order
19         Arrays.sort(IntArray, Collections.reverseOrder());
20
21         //print sorted array
22         System.out.printf("\n\nSorted Array: %s",
23             Arrays.toString(IntArray));
24     }
25 }
26
27
```

Console Output:

```
<terminated> ArraySortDecendingOrder [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\java.exe
Original Array: [52, 45, 32, 64, 12, 87, 78, 98, 23, 7]
Sorted Array: [98, 87, 78, 64, 52, 45, 32, 23, 12, 7]
```

13) Find Unique Elements in Array

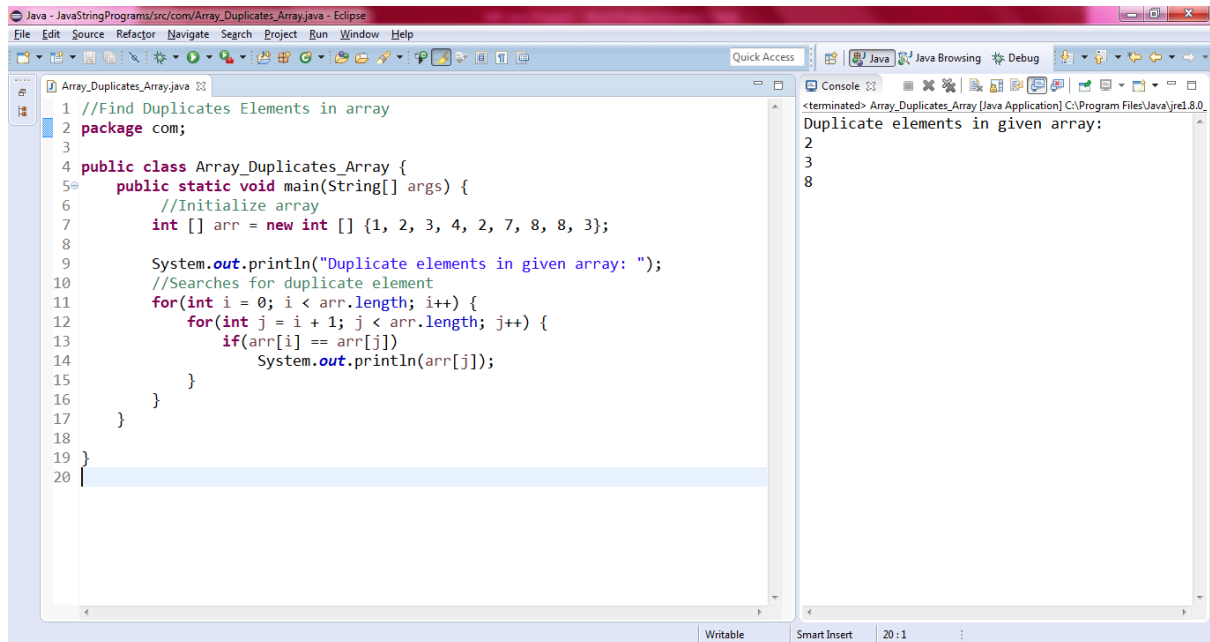
The screenshot shows the Eclipse IDE with a Java project named 'JavaStringPrograms'. The editor displays the file 'Array_UniqueElements.java'. The code defines a class 'Array_UniqueElements' with a 'main' method. It defines an integer array 'arr' with values {1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 5, 5, 6, 7, 8}. The program uses nested loops to find unique elements. It counts the frequency of each element and prints the unique elements: 6, 7, 8. The console output shows the unique elements: 6 7 8.

```
1 //unique numbers in array java
2 package com;
3 public class Array_UniqueElements
4 {
5     public static void main(String[] args) {
6         int [] arr = {1,1,1,2,2,2,3,3,3,4,4,5,5,6,7,8,};
7
8         for (int j= 0; j < arr.length; j++)
9         {
10             int count1 = 0; // to find out how many time value is appeared in the
11             for (int i = 0; i < arr.length; i++)
12             {
13                 if(arr[i] == arr[j] )
14                 {
15                     count1++;
16                 }
17             }
18             if(count1 == 1 )
19             {
20                 System.out.print(arr[j]+" "); // 6 7 8 unique
21             }
22         }
23     }
24 }
25
26
```

Console Output:

```
<terminated> Array_UniqueElements [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\java.exe
6 7 8
```

14) Find Duplicate Element in Array

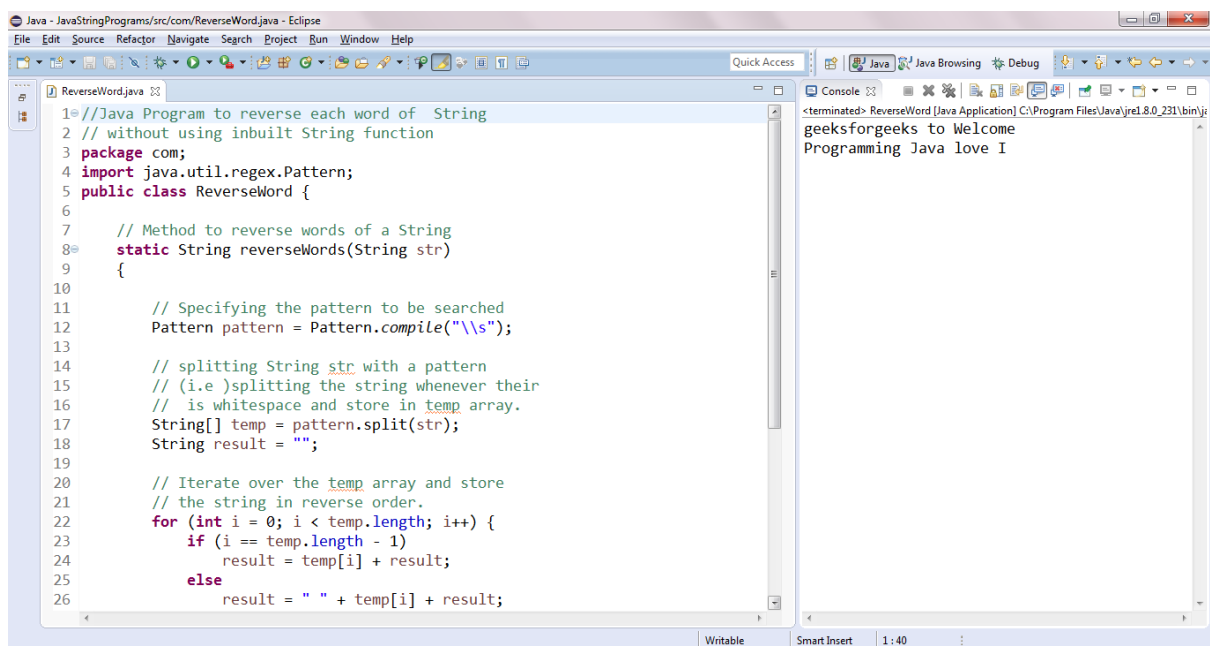


```
1 //Find Duplicates Elements in array
2 package com;
3
4 public class Array_Duplicates_Array {
5     public static void main(String[] args) {
6         //Initialize array
7         int [] arr = new int [] {1, 2, 3, 4, 2, 7, 8, 8, 3};
8
9         System.out.println("Duplicate elements in given array: ");
10        //Searches for duplicate element
11        for(int i = 0; i < arr.length; i++) {
12            for(int j = i + 1; j < arr.length; j++) {
13                if(arr[i] == arr[j])
14                    System.out.println(arr[j]);
15            }
16        }
17    }
18 }
19 }
20 }
```

Console Output:

```
<terminated> Array_Duplicates_Array [Java Application] C:\Program Files\Java\jre1.8.0_
Duplicate elements in given array:
2
3
8
```

15) Java Program for Reverse Each word of String



```
1 //Java Program to reverse each word of String
2 // without using inbuilt String function
3 package com;
4 import java.util.regex.Pattern;
5 public class ReverseWord {
6
7     // Method to reverse words of a String
8     static String reverseWords(String str)
9     {
10
11         // Specifying the pattern to be searched
12         Pattern pattern = Pattern.compile("\\s");
13
14         // splitting String str with a pattern
15         // (i.e) splitting the string whenever their
16         // is whitespace and store in temp array.
17         String[] temp = pattern.split(str);
18         String result = "";
19
20         // Iterate over the temp array and store
21         // the string in reverse order.
22         for (int i = 0; i < temp.length; i++) {
23             if (i == temp.length - 1)
24                 result = temp[i] + result;
25             else
26                 result = " " + temp[i] + result;
27         }
28     }
29 }
```

Console Output:

```
<terminated> ReverseWord [Java Application] C:\Program Files\Java\jre1.8.0_231\bin\j
geeksforgEEKS to WelcOME
Programming Java love I
```

