

Self Practice - Strings

1. Given a string, return the string made of its first two chars, so the String "Hello" yields "He". If the string is shorter than length 2, return whatever there is, so "X" yields "X", and the empty string "" yields the empty string "".

Analysis:

1. Get the first two characters of the string.
2. If the string is empty, return the empty string.
3. If the string is less than 2 characters, return the string as is.
4. If the string is more than 2 characters, return the first two characters.

Program: FirstTwoChars.java

Code

```
package com.self_practice;

import java.util.Scanner;

public class FirstTwoChars {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the String: ");
        String str = sc.nextLine();
        System.out.println(firstTwo(str));
    }

    public static String firstTwo(String str) {
        if (str.length() < 2) {
            return str;
        } else {
            return str.substring(0, 2);
        }
    }
}
```

```
}  
}
```

Output

```
Enter the String: Hello  
He
```

2. Given 2 strings, a and b, return a string of the form short+long+short, with the shorter string on the outside and the longer string on the inside. The strings will not be the same length, but they may be empty (length 0)

Analysis:

1. Get the length of both strings.
2. If the first string is smaller than the second string, return the first string + second string + first string.
3. If the second string is smaller than the first string, return the second string + first string + second string.

Program: ShortLongShort.java

Code

```
package com.self_practice;  
  
import java.util.Scanner;  
  
public class ShortLongShort {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.print("Enter the first String: ");  
        String a = sc.nextLine();  
        System.out.print("Enter the second String: ");  
        String b = sc.nextLine();  
    }  
}
```

```

        System.out.println(shortLongShort(a, b));
    }

    public static String shortLongShort(String a, String b) {
        if (a.length() < b.length()) {
            return a + b + a;
        } else {
            return b + a + b;
        }
    }
}

```

Output

```

Enter the first String:  Shabari
Enter the second String: K S
K S Shabari K S

```

3. Given a string of any length, return a new string where the last 2 chars, if present, are swapped, so "coding" yields "codign".

Analysis:

1. Get the length of the string.
2. If the length of the string is less than 2, return the string as is.
3. If the length of the string is more than 2, swap the last two characters.

Program: SwapLastTwoChars.java

Code

```

package com.self_practice;

import java.util.Scanner;

public class SwapLastTwoChars {

```

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter the String: ");
    String str = sc.nextLine();
    System.out.println(swapLastTwoChars(str));
}

public static String swapLastTwoChars(String str) {
    if (str.length() < 2) {
        return str;
    } else {
        return str.substring(0, str.length() - 2)
            + str.charAt(str.length() - 1) +
            str.charAt(str.length() - 2);
    }
}
}
```

Output

```
Enter the String: coding
codign
```

4. A string S is passed as the input. S can contain alphabets, numbers and special characters. The program must print only the alphabets in S.

Input:

```
abcd_5ef8!xyz
```

Output:

```
abcdefxyz
```

Analysis:

1. Get the input string.
2. Use the regular expression to match only the alphabets.
3. Print the matched alphabets.

Program: PrintAlphabets.java

Code

```
package com.self_practice;

import java.util.Scanner;

public class PrintAlphabets {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the string: ");
        String userInput = sc.nextLine();
        System.out.print(printAlphabets(userInput));
    }
    public static String printAlphabets(String str) {
        return str.replaceAll("[^a-zA-Z]", "");
    }
}
```

Output

```
Enter the String: abcd_5ef8!xyz
abcdefxyz
```

5. Write a program to sort a set of names stored in an array in alphabetical order.

Analysis:

1. Get the input names.
2. Sort the names in alphabetical order.
3. Print the sorted names.

Program: SortNames.java

Code

```
package com.self_practice;

import java.util.Arrays;

public class SortNames {
    public static void main(String[] args) {
        String[] names = {"Subasri", "Shabari", "Sathyaram", "Sowandharya" };
        Arrays.sort(names);
        for (String name : names) {
            System.out.println(name);
        }
    }
}
```

Output

```
Sathyaram
Shabari
Sowandharya
Subasri
```

6. Write a Java program to count the number of occurrences of a character in a String.

Analysis:

1. Get the input string.
2. Get the character to be counted.
3. Initialize the count to 0.
4. Iterate through the string and check if the character is present.
5. If present, increment the count.
6. Print the count.

Program: CountCharacter.java

Code

```
package com.self_practice;

import java.util.Scanner;

public class CountCharacter {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the string: ");
        String str = sc.nextLine();
        System.out.print("Enter the character: ");
        char ch = sc.next().charAt(0);
        int count=0;
        for(char c : str.toCharArray()){
            if(ch == c){
                count++;
            }
        }
        System.out.println("The Occurence of the "+ch+" is "+ count);
    }
}
```

Output

```
Enter the string: Shabari
Enter the character: a
The Occurence of the a is 2
```

7. Given a string, return a string length 2 made of its first 2 character. If the string length is less than 2, use '@' for the missing chars.

Analysis:

1. Get the input string.

2. If the length of the string is less than 2, return the string + '@'.
3. If the length of the string is more than 2, return the first two characters.

Program: `FirstTwoCharsWithAt.java`

Code

```
package com.self_practice;

import java.util.Scanner;

public class FirstTwoCharsWithAt {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the string: ");
        String str = sc.nextLine();
        if(str.length() < 2){
            System.out.print(str);
            for(int i=0; i<2; i++){
                System.out.print("@");
            }
        }
        else{
            System.out.print(str.substring(0,2));
        }
    }
}
```



Output

```
Enter the string:
a
a@
```

8. Write a Java program to create a new string repeating every character twice of a given string

Analysis:

1. Get the input string.
2. Initialize an empty string.
3. Iterate through the string and append each character twice to the new string.
4. Print the new string.

Program: RepeatEveryCharacterTwice.java

Code

```
package com.self_practice;

import java.util.Scanner;

public class RepeatEveryCharacterTwice {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the string: ");
        String str = sc.nextLine();
        String newStr = "";
        for (int i = 0; i < str.length(); i++) {
            newStr += str.charAt(i);
            newStr += str.charAt(i);
        }
        System.out.println(newStr);
    }
}
```

Output

```
Enter the string: shabari
sshhaabbaarrii
```

9. Write a Java program to return the sum of the digits present in the given string. If there is no digits the sum

return is 0.

Analysis:

1. Get the input string.
2. Initialize the sum to 0.
3. Iterate through the string and check if the character is a digit.
4. If it is a digit, convert it to an integer and add it to the sum.
5. Print the sum.

Program: SumOfDigits.java

Code

```
package com.self_practice;

import java.util.Scanner;

public class SumOfDigits {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the string: ");
        String str = sc.nextLine();
        int sum = 0;
        for (char c : str.toCharArray()) {
            if (Character.isDigit(c)) {
                sum += Integer.parseInt(String.valueOf(c));
            }
        }
        System.out.println(sum);
    }
}
```

Output

```
Enter the string: shabari123
6
```

10. Write a Java program to print after removing duplicates from a given string.

Analysis:

1. Get the input string.
2. Initialize an empty string.
3. Iterate through the string and check if the character is already present in the new string.
4. If not present, append the character to the new string.
5. Print the new string.

Program: RemoveDuplicates.java

Code

```
package com.self_practice;

import java.util.Scanner;

public class RemoveDuplicates {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the string: ");
        String str = sc.nextLine();
        String newStr = "";
        for (char c : str.toCharArray()) {
            if (newStr.indexOf(c) == -1) {
                newStr += c;
            }
        }
        System.out.println(newStr);
    }
}
```

Output

```
Enter the string: shabari
shabri
```

