

**Week 1 - S1 - Lab Problem**

**Name: Ramesh Harisabapathi Chettiar**

**Roll Number: RA2411030010263**

**Course: Networking and Communications**

**Semester: 3**

**Date of Submission: 19/08/2025**

## Lab Practice Programs (Any Six)

1. Write a program to find and return the length of a string without using the length() method

Hint =>

a. Take user input using the Scanner next() method

b. Create a method to find and return a string's length without using the built-in length() method. The logic for this is to use the infinite loop to count each character till the charAt() method throws a runtime exception, handles the exception, and then return the count

c. The main function calls the user-defined method as well as the built-in length() method and displays the result

```
1  //Write a program to find and return the length of a string without using the length() method
2  import java.util.Scanner;
3
4  public class StringLength{
5      public static int Length(String text) {
6          int count = 0;
7          int i = 0;
8          try {
9              while (i >= 0) {
10                 if (Character.isLetterOrDigit(text.charAt(i))) {
11                     count++;
12                 }
13                 i++;
14             }
15         } catch (RuntimeException e) {
16             System.out.println("Caught RuntimeException: " + e.getMessage());
17         }
18         return count+1;
19     }
20
21     public static void main(String args[]){
22         Scanner input = new Scanner(System.in);
23
24         System.out.println("Enter String:");
25         String str = input.nextLine();
26
27
28         //Function calling
29         int length = Length(str);
30         System.out.println("User Defined Function-->"+length);
31         System.out.println("Inbuilt function-->"+str.length());
32         if (length == str.length()){
33             System.out.println("Both the Inbuilt function and the user defined function return the same length of the String.");
34         }
35
36
37         input.close();
38     }
39 }
```

## 5. Write a program to find vowels and consonants in a string and display the count of Vowels and Consonants in the string

Hint =>

a. Create a method to check if the character is a vowel or consonant and return the result.

The logic used here is as follows:

i. Convert the character to lowercase if it is an uppercase letter using the ASCII values of the characters

ii. Check if the character is a vowel or consonant and return Vowel, Consonant, or Not a Letter

b. Create a Method to Method to find vowels and consonants in a string using charAt() method and finally return the count of vowels and consonants in an array

c. Finally, the main function takes user inputs, calls the user-defined methods, and displays the result.

```
1  /*Write a program to find vowels and consonants in a string and display the count of Vowels
2  and Consonants in the string
3  Hint =>
4  a. Create a method to check if the character is a vowel or consonant and return the result.
5  The logic used here is as follows:
6
7  i. Convert the character to lowercase if it is an uppercase letter using the ASCII values
8  of the characters
9
10 ii. Check if the character is a vowel or consonant and return Vowel, Consonant, or Not
11 a Letter
12 b. Create a Method to Method to find vowels and consonants in a string using charAt()
13 method and finally return the count of vowels and consonants in an array
14 c. Finally, the main function takes user inputs, calls the user-defined methods, and displays
15 the result. */
16
17 import java.util.Scanner;
18
19 public class VowelConsonant {
20     public static int Length(String text) {
21         int count = 0;
22         try {
23             while (true) { // Keep going until we reach the end
24                 text.charAt(count); // Will throw exception if out of bounds
25                 count++;
26             }
27         } catch (IndexOutOfBoundsException e) {
28             // End of string reached
29         }
30         return count;
31     }
32 }
```

```

34 // Checks and print whether each character is vowel, consonant, or non-letter
35 public static void isVowelConsonant(String text) {
36     String newText = text.toLowerCase();
37
38     for (int i = 0; i < Length(newText); i++) {
39         char c = newText.charAt(i);
40
41         if (c >= 'a' && c <= 'z') { // Only letters
42             if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u') {
43                 System.out.println(c + " is a vowel.");
44             }
45
46             else {
47                 System.out.println(c + " is a consonant.");
48             }
49         }
50
51         else {
52             System.out.println(c + " is not a letter.");
53         }
54     }
55 }
56

```

```

57 // Count vowels and consonants
58 public static int[] VowelConsonants(String text) {
59     int vowels = 0, consonants = 0;
60
61     for (int i = 0; i < Length(text); i++) {
62         char c = Character.toLowerCase(text.charAt(i));
63
64         if (c >= 'a' && c <= 'z') { // Only letters
65             if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u') {
66                 vowels++;
67             }
68
69             else {
70                 consonants++;
71             }
72         }
73     }
74
75     return new int[]{vowels, consonants};
76 }
77

```

```

78 public static void main(String[] args) {
79     Scanner input = new Scanner(System.in);
80
81     System.out.println("Enter a sentence:");
82     String sentence = input.nextLine();
83
84     isVowelConsonant(sentence);
85
86     int[] counts = VowelConsonants(sentence);
87     System.out.println("\nSentence: " + sentence);
88     System.out.println("Number of vowels: " + counts[0]);
89     System.out.println("Number of consonants: " + counts[1]);
90
91     input.close();
92 }
93 }
94

```

6. Write a program to find vowels and consonants in a string and display the character type -

Vowel, Consonant, or Not a Letter

Hint =>

a. Create a method to check if the character is a vowel or consonant and return the result.

The logic used here is as follows:

i. Convert the character to lowercase if it is an uppercase letter using the ASCII values of the characters

ii. Check if the character is a vowel or consonant and return Vowel, Consonant, or Not a Letter

b. Create a Method to find vowels and consonants in a string using charAt() method and return the character and vowel or consonant in a 2D array

c. Create a Method to display the 2D Array of Strings in a Tabular Format

d. Finally, the main function takes user inputs, calls the user-defined methods, and displays the result.

```
1  /*Write a program to find vowels and consonants in a string and display the character type -
2  Vowel, Consonant, or Not a Letter
3  Hint =>
4  a. Create a method to check if the character is a vowel or consonant and return the result.
5  The logic used here is as follows:
6  i. Convert the character to lowercase if it is an uppercase letter using the ASCII values
7  of the characters
8  ii. Check if the character is a vowel or consonant and return Vowel, Consonant, or Not
9  a Letter
10 b. Create a Method to find vowels and consonants in a string using charAt() method and
11 return the character and vowel or consonant in a 2D array
12 c. Create a Method to display the 2D Array of Strings in a Tabular Format
13 d. Finally, the main function takes user inputs, calls the user-defined methods, and displays
14 the result. */
15
16 import java.util.Scanner;
17
18 public class CharacterType {
19
20     // Custom method to calculate string length without using .length()
21     public static int Length(String text) {
22         int count = 0;
23         try {
24             while (true) {
25                 text.charAt(count); // Throws exception when index is out of bounds
26                 count++;
27             }
28         } catch (IndexOutOfBoundsException e) {
29             // End of string reached
30         }
31         return count;
32     }
33 }
```

```

34 // Checks and prints whether each character is a vowel, consonant, or non-letter
35 public static void isVowelConsonant(String text) {
36     String newText = text.toLowerCase();
37
38     for (int i = 0; i < Length(newText); i++) {
39         char c = newText.charAt(i);
40
41         if (c >= 'a' && c <= 'z') {
42             if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u') {
43                 System.out.println(c + " is a vowel.");
44             } else {
45                 System.out.println(c + " is a consonant.");
46             }
47         } else {
48             System.out.println(c + " is not a letter.");
49         }
50     }
51 }
52
53 public static void main(String[] args) {
54     Scanner input = new Scanner(System.in);
55
56     System.out.println("Enter a sentence:");
57     String sentence = input.nextLine();
58
59     isVowelConsonant(sentence);
60
61     input.close();
62 }
63 }
64
65

```

7. Write a program to trim the leading and trailing spaces from a string using the `charAt()` method

Hint =>

a. Create a method to trim the leading and trailing spaces from a string using the `charAt()` method. Inside the method run a couple of loops to trim leading and trailing spaces and determine the starting and ending points with no spaces. Return the start point and end point in an array

b. Write a method to create a substring from a string using the `charAt()` method with the string, start, and end index as the parameters

c. Write a method to compare two strings using the `charAt()` method and return a Boolean result

d. The main function calls the user-defined trim and substring methods to get the text after

trimming the leading and trailing spaces. Post that use the String built-in method `trim()` to trim spaces and compare the two strings. And finally display the result

```
1  /*7. Write a program to trim the leading and trailing spaces from a string using the charAt()
2  method
3  Hint =>
4  a. Create a method to trim the leading and trailing spaces from a string using the charAt()
5  method. Inside the method run a couple of loops to trim leading and trailing spaces and
6  determine the starting and ending points with no spaces. Return the start point and end
7  point in an array
8  b. Write a method to create a substring from a string using the charAt() method with the
9  string, start, and end index as the parameters
10 c. Write a method to compare two strings using the charAt() method and return a boolean
11 result
12 d. The main function calls the user-defined trim and substring methods to get the text after
13 trimming the leading and trailing spaces. Post that use the String built-in method trim()
14 to trim spaces and compare the two strings. And finally display the result */
15
16 import java.util.Scanner;
17
18 public class CustomTrim{
19     // Custom method to calculate string length without using .length()
20     public static int Length(String text) {
21         int count = 0;
22         try {
23             while (true) {
24                 text.charAt(count); // Throws exception when index is out of bounds
25                 count++;
26             }
27         } catch (IndexOutOfBoundsException e) {
28             // End of string reached
29         }
30         return count;
31     }
32 }
```

```

33     public static String trimFunction(String text){
34         StringBuilder sb = new StringBuilder();
35         int count1 = 0, count2 = 0;
36         int i = 0;
37         while(i < Length(text) ){
38             if (text.charAt(i) == ' '){
39                 count1 ++;
40             }
41             else{
42                 break;
43             }
44             i++;
45         }
46
47         i = Length(text) - 1;
48         while(i >= 0){
49             if (text.charAt(i) == ' '){
50                 count2 ++;
51             }
52             else{
53                 break;
54             }
55             i--;
56         }
57
58         for (int j = count1; j < Length(text) - count2 ; j++){
59             sb.append(text.charAt(j));
60         }
61
62         String returnString = sb.toString();
63         return returnString;
64     }

```

```

66     public static void main(String args[]){
67         Scanner input = new Scanner(System.in);
68
69         System.out.println("Enter sentence-->");
70         String sentence = input.nextLine();
71
72         System.out.println("After Trim-->" + trimFunction(sentence));
73
74         input.close();
75     }
76 }

```



**8. Write a program to take user input for the age of all 10 students in a class and check whether the student can vote depending on his/her age is greater or equal to 18.**

**Hint =>**

- a. Create a method to define the random 2-digit age of several students provided as method parameters and return a 1D array of ages of n students**
- b. Create a method that takes an array of age as a parameter and returns a 2D String array of age and a boolean true or false to indicate can and cannot vote. Inside the method firstly validate the age for a negative number, if a negative cannot vote. For valid age check for age is 18 or above to set true to indicate can vote.**
- c. Create a method to display the 2D array in a tabular format.**
- d. Finally, the main function takes user inputs, calls the user-defined methods, and displays the result.**

```

1  /*Write a program to take user input for the age of all 10 students in a class and check
2  whether the student can vote depending on his/her age is greater or equal to 18.
3  Hint =>
4  a. Create a method to define the random 2-digit age of several students provided as
5  method parameters and return a 1D array of ages of n students
6  b. Create a method that takes an array of age as a parameter and returns a 2D String array
7  of age and a boolean true or false to indicate can and cannot vote. Inside the method
8  firstly validate the age for a negative number, if a negative cannot vote. For valid age
9  check for age is 18 or above to set true to indicate can vote.
10 c. Create a method to display the 2D array in a tabular format.
11 d. Finally, the main function takes user inputs, calls the user-defined methods, and displays
12 the result. */
13
14 import java.util.Scanner;
15
16 public class StudentVoter {
17     static Scanner input = new Scanner(System.in);
18
19     // Function 1: Take ages
20     public static int[] age(int n) {
21         int[] array = new int[n];
22         for (int i = 0; i < n; i++) {
23             System.out.print("Enter age (10-100) --> ");
24             int age = input.nextInt();
25             if (age < 10 || age > 100) {
26                 System.out.println("Invalid age! Enter again.");
27                 i--; // repeat for same index
28             } else {
29                 array[i] = age;
30             }
31         }
32         return array;
33     }
34
35     // Function 2: Check voting eligibility
36     public static boolean[] Array(int[] ages) {
37         boolean[] canVote = new boolean[ages.length];
38         for (int i = 0; i < ages.length; i++) {
39             canVote[i] = (ages[i] >= 18); // true if >= 18
40         }
41         return canVote;
42     }
43
44     // Function 3: Display details
45     public static void DisplayArray(int[] ages, boolean[] votes) {
46         System.out.println("\n-----DETAILS-----");
47         System.out.println("Person\t\tAge\t\tCanVote?");
48         for (int i = 0; i < ages.length; i++) {
49             System.out.println((i + 1) + "\t\t" + ages[i] + "\t\t" + votes[i]);
50         }
51     }
52
53     // Main
54     public static void main(String args[]) {
55         System.out.print("Enter Number of People --> ");
56         int n = input.nextInt();
57
58         int[] PersonAges = age(n);
59         boolean[] voteChecker = Array(PersonAges);
60         DisplayArray(PersonAges, voteChecker);
61
62         input.close();
63     }
64 }

```

**9. Rock-Paper-Scissors is a game played between a minimum of two players. Each player can**

**choose either rock, paper, or scissors. Here the game is played between a user and a computer. Based on the rules, either a player or a computer will win. Show the stats of player and computer win in a tabular format across multiple games. Also, show the winning**

**percentage between the player and the computer.**

**Hint =>**

**a. The rule is: rock-scissors: rock will win (rock crushes scissors); rock-paper: paper wins (paper covers rock); scissors-paper: scissors win (scissors cuts paper)**

**b. Create a Method to find the Computer Choice using the Math.random**

**c. Create a Method to find the winner between the user and the computer**

**d. Create a Method to find the average and percentage of wins for the user and the computer and return a String 2D array**

**e. Create a Method to display the results of every game and also display the average and percentage wins**

**f. In the main take user input for the number of games and call methods to display results**

```

1  /*Rock-Paper-Scissors is a game played between a minimum of two players. Each player can
2  choose either rock, paper, or scissors. Here the game is played between a user and a
3  computer. Based on the rules, either a player or a computer will win. Show the stats of
4  player and computer win in a tabular format across multiple games. Also, show the winning
5  percentage between the player and the computer.
6  Hint =>
7  a. The rule is: rock-scissors: rock will win (rock crushes scissors); rock-paper: paper wins
8  (paper covers rock); scissors-paper: scissors win (scissors cuts paper)
9  b. Create a Method to find the Computer Choice using the Math.random
10 c. Create a Method to find the winner between the user and the computer
11 d. Create a Method to find the average and percentage of wins for the user and the
12 computer and return a String 2D array
13 e. Create a Method to display the results of every game and also display the average and
14 percentage wins
15 f. In the main take user input for the number of games and call methods to display results */
16
17 import java.util.Scanner;
18 import java.util.Random;
19
20 public class RockPaperScissors {
21
22     // Function to choose on behalf of the computer
23     public static String computerChoice() {
24         Random random = new Random();
25         int c = 1 + random.nextInt(3); // random number (1-3)
26         if (c == 1) return "rock";
27         else if (c == 2) return "paper";
28         else return "scissors";
29     }
30
31     // Function to find out the Winner between User and Computer
32     public static String winner(String cChoice, String uChoice) {
33         if (cChoice.equals(uChoice)) {
34             return "Tie";
35         }
36         if ((cChoice.equals("rock") && uChoice.equals("scissors")) ||
37             (cChoice.equals("scissors") && uChoice.equals("paper")) ||
38             (cChoice.equals("paper") && uChoice.equals("rock"))) {
39             return "Computer";
40         } else {
41             return "User";
42         }
43     }
44
45     // Function to calculate average and win percentage
46     public static String[][] statistics(int userWins, int compWins, int games) {
47         String[][] stats = new String[2][3];
48
49         double userPercent = ((double) userWins / games) * 100;
50         double compPercent = ((double) compWins / games) * 100;
51
52         stats[0][0] = "User";
53         stats[0][1] = String.valueOf(userWins);
54         stats[0][2] = String.format("%.2f", userPercent);
55
56         stats[1][0] = "Computer";
57         stats[1][1] = String.valueOf(compWins);
58         stats[1][2] = String.format("%.2f", compPercent);
59
60         return stats;
61     }
62 }

```

```

63 // Function to display results
64 public static void displayResults(String[][] stats) {
65     System.out.println("\n----- Final Statistics -----");
66     System.out.println("Player\t\tWins\t\tWin %");
67
68     for (int i = 0; i < stats.length; i++) {
69         System.out.println(stats[i][0] + "\t\t" + stats[i][1] + "\t\t" + stats[i][2]);
70     }
71 }
72
73 public static void main(String args[]) {
74     Scanner input = new Scanner(System.in);
75
76     System.out.print("Enter number of games to play: ");
77     int n = input.nextInt();
78
79     int userWins = 0, compWins = 0, ties = 0;
80
81     for (int i = 1; i <= n; i++) {
82         System.out.print("\nGame " + i + " - Enter your choice (rock(0)/paper(1)/scissors(any other num)");
83         String uChoice = input.next().toLowerCase();
84         String cChoice = computerChoice();
85
86         System.out.println("Computer chose: " + cChoice);
87
88         String result = winner(cChoice, uChoice);
89
90         if (result.equals("User")) userWins++;
91         else if (result.equals("Computer")) compWins++;
92         else ties++;
93
94         System.out.println("Winner: " + result);
95     }
96
97     String[][] stats = statistics(userWins, compWins, n);
98     displayResults(stats);
99
100     System.out.println("\nTies: " + ties);
101
102     input.close();
103 }
104 }
105
106
107

```

## OUTPUTS

1.

```
Enter String:
Ramesh Harisabapathi Chettia
Caught RuntimeException: Index 28 out of bounds for length 28
User Defined Function-->27
Inbuilt function-->28
```

```
Enter a sentence:
Java STEP Programming
j is a consonant.
a is a vowel.
v is a consonant.
a is a vowel.
  is not a letter.
s is a consonant.
t is a consonant.
e is a vowel.
p is a consonant.
  is not a letter.
p is a consonant.
r is a consonant.
o is a vowel.
g is a consonant.
r is a consonant.
a is a vowel.
m is a consonant.
m is a consonant.
i is a vowel.
n is a consonant.
g is a consonant.

Sentence: Java STEP Programming
Number of vowels: 6
Number of consonants: 13
```

5.

6.

```
Enter a sentence:  
I love to learn Java as it is fun and easy to learn.  
i is a vowel.  
  is not a letter.  
l is a consonant.  
o is a vowel.  
v is a consonant.  
e is a vowel.  
  is not a letter.  
t is a consonant.  
o is a vowel.  
  is not a letter.  
l is a consonant.  
e is a vowel.  
a is a vowel.  
r is a consonant.  
n is a consonant.  
  is not a letter.  
j is a consonant.  
a is a vowel.  
v is a consonant.  
a is a vowel.  
  is not a letter.  
a is a vowel.  
s is a consonant.  
  is not a letter.  
i is a vowel.  
t is a consonant.  
  is not a letter.  
i is a vowel.  
s is a consonant.  
  is not a letter.
```

```
f is a consonant.  
u is a vowel.  
n is a consonant.  
  is not a letter.  
a is a vowel.  
n is a consonant.  
d is a consonant.  
  is not a letter.  
e is a vowel.  
a is a vowel.  
s is a consonant.  
y is a consonant.  
  is not a letter.  
t is a consonant.  
o is a vowel.  
  is not a letter.  
l is a consonant.  
e is a vowel.  
a is a vowel.  
r is a consonant.  
n is a consonant.  
. is not a letter.
```

7.

```
Enter sentence-->
```

```
I Love Java.It is fun and easy to learn
```

```
After Trim-->I Love Java.It is fun and easy to learn
```



```

Enter Number of People --> 5
Enter age (10-100) --> 20
Enter age (10-100) --> 833
Invalid age! Enter again.
Enter age (10-100) --> 92
Enter age (10-100) --> 83
Enter age (10-100) --> 92
Enter age (10-100) --> 74

```

```

-----DETAILS-----
Person      1Age      CanVote?
1           20      true
2           92      true
3           83      true
4           92      true
5           74      true

```

8.

9.

```

Enter number of games to play: 5

Game 1 - Enter your choice (rock(0)/paper(1)/scissors(any other number)): 0
Computer chose: scissors
Winner: User

Game 2 - Enter your choice (rock(0)/paper(1)/scissors(any other number)): 8
Computer chose: scissors
Winner: User

Game 3 - Enter your choice (rock(0)/paper(1)/scissors(any other number)): 93
Computer chose: rock
Winner: User

Game 4 - Enter your choice (rock(0)/paper(1)/scissors(any other number)): 72
Computer chose: scissors
Winner: User

Game 5 - Enter your choice (rock(0)/paper(1)/scissors(any other number)): 94
Computer chose: paper
Winner: User

----- Final Statistics -----
Player      Wins      Win %
User         5         100.00
Computer     0          0.00

Ties: 0

```