

Level 1 Practice Programs

1. Write a program to compare two strings using the `charAt()` method and check the result with the built-in String `equals()` method

Hint =>

- a. Take user input using the `Scanner next()` method for 2 String variables
- b. Write a method to compare two strings using the `charAt()` method and return a boolean result
- c. Use the String Built-In method to check if the results are the same and display the result

```
CODE- import java.util.Scanner;

public class StringCompare {

    public static boolean compareStringsUsingCharAt(String str1,
String str2) {

        if (str1.length() != str2.length()) {
            return false;
        }

        for (int i = 0; i < str1.length(); i++) {
            if (str1.charAt(i) != str2.charAt(i)) {
                return false;
            }
        }
        return true;
    }

    public static void main(String[] args) {
```

```

Scanner scanner = new Scanner(System.in);

        System.out.print("Enter first string: ");
String string1 = scanner.next();

        System.out.print("Enter second string: ");
String string2 = scanner.next();

        boolean charAtResult =
compareStringsUsingCharAt(string1, string2);

        boolean equalsResult = string1.equals(string2);

        System.out.println("Result using charAt() comparison: "
+ charAtResult);

        System.out.println("Result using equals() method: " +
equalsResult);

        if (charAtResult == equalsResult) {
            System.out.println("Both methods give the same result.");
        } else {
            System.out.println("Methods give different results.");
        }

        scanner.close();
    }
}

```

2. Write a program to create a substring from a String using the **charAt()** method. Also, use the String built-in method **substring()** to find the substring of the text. Finally Compare the the two strings and display the results

Hint =>

- a. Take user input using the **Scanner next()** method to take the String variable and also the start and the end index to get the substring from the given text
- 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. Use the String built-in method substring() to get the substring and compare the two strings. And finally display the result

CODE– `import java.util.Scanner;`

`public class SubstringCompare {`

`public static String createSubstringUsingCharAt(String text, int start, int end) {`

`StringBuilder result = new StringBuilder();`

`for (int i = start; i < end && i < text.length(); i++) {`

`result.append(text.charAt(i));`

`}`

`return result.toString();`

`}`

`public static boolean compareStringsUsingCharAt(String str1, String str2) {`

`if (str1.length() != str2.length()) {`

`return false;`

`}`

`for (int i = 0; i < str1.length(); i++) {`

`if (str1.charAt(i) != str2.charAt(i)) {`

`return false;`

`}`

`}`

```

        return true;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a string: ");
        String input = scanner.next();

        System.out.print("Enter start index: ");
        int start = scanner.nextInt();

        System.out.print("Enter end index: ");
        int end = scanner.nextInt();

        String customSubstring = createSubstringUsingCharAt(input,
start, end);
        String builtInSubstring = "";

        if (start >= 0 && end <= input.length() && start < end) {
            builtInSubstring = input.substring(start, end);
        } else {
            System.out.println("Invalid indices for substring().");
            scanner.close();
            return;
        }
    }

```



```

        boolean isEqual = compareStringsUsingCharAt(customSubstring,
builtInSubstring);

        System.out.println("Substring using charAt(): " +
customSubstring);
        System.out.println("Substring using substring(): " +
builtInSubstring);
        System.out.println("Are both substrings equal? " + isEqual);

        scanner.close();
    }
}

```

3. Write a program to return all the characters in a string using the user-defined method, compare the result with the String built-in toCharArray() method, and display the result

Hint =>

- Take user input using the **Scanner next()** method to take the text into a String variable
- Write a method to return the characters in a string without using the **toCharArray()**
- Write a method to compare two string arrays and return a boolean result
- In the main() call the user-defined method and the String built-in toCharArray() method, compare the 2 arrays, and finally display the result

CODE- import java.util.Scanner;

public class CharArrayCompare {

```

    public static char[] getCharsFromString(String text) {
        char[] result = new char[text.length()];
        for (int i = 0; i < text.length(); i++) {
            result[i] = text.charAt(i);
        }
    }
}

```

```
    }  
    return result;  
}
```

```
    public static boolean compareCharArrays(char[] array1, char[]  
array2) {  
        if (array1.length != array2.length) {  
            return false;  
        }  
  
        for (int i = 0; i < array1.length; i++) {  
            if (array1[i] != array2[i]) {  
                return false;  
            }  
        }  
  
        return true;  
}
```

```
public static void main(String[] args) {  
    Scanner scanner = new Scanner(System.in);  
  
    System.out.print("Enter a string: ");  
    String input = scanner.next();  
  
    char[] customChars = getCharsFromString(input);  
  
    char[] builtInChars = input.toCharArray();
```



```

        boolean isEqual = compareCharArrays(customChars,
builtInChars);

        System.out.println("Characters using custom method:");
        for (char c : customChars) {
            System.out.print(c + " ");
        }

        System.out.println("\nCharacters using toCharArray():");
        for (char c : builtInChars) {
            System.out.print(c + " ");
        }

        System.out.println("\nAre both character arrays equal? " +
isEqual);

        scanner.close();
    }
}

```

4. Write a program to demonstrate NullPointerException.

Hint =>

- a. Write a Method to generate the Exception. Here define the variable text and initialize it to null. Then call one of the String Method to generate the exception
- e. Write the Method to demonstrate **NullPointerException**. Here define the variable text and initialize it to null. Then write try catch block for handling the Exception while accessing one of the **String** method
- b. From the main Firstly call the method to generate the Exception then refactor the code to call the method to handle the RuntimeException

```

CODE- public class NullPointerExceptionDemo {

    public static void generateException() {
        String text = null;

        System.out.println("Length of text: " + text.length());
    }

    public static void handleException() {
        String text = null;
        try {

            System.out.println("Length of text: " + text.length());
        } catch (NullPointerException e) {

            System.out.println("NullPointerException caught: The string is
null.");
        }
    }

    public static void main(String[] args) {

        handleException();
    }
}

```

5. Write a program to demonstrate ***StringIndexOutOfBoundsException***

Hint =>

- a. Define a variable of type String and take user input to assign a value
- b. Write a Method to generate the Exception. Access the index using `charAt()` beyond the length of the String. This will generate a runtime exception and abruptly stop the program.
- c. Write the Method to demonstrate ***StringIndexOutOfBoundsException***. Access the index using ***charAt()*** beyond the length of the String. Then write try catch block for Exception while accessing the String method
- d. From the main Firstly call the method to generate the Exception then call the method to handle the RuntimeException

CODE- `import java.util.Scanner;`

`public class StringIndexDemo {`

`public static void generateStringIndexOutOfBoundsException(String text) {`

`System.out.println("Character at invalid index: " +
text.charAt(text.length()));
 }`

`public static void handleStringIndexOutOfBoundsException(String text) {`

`try {

 System.out.println("Character at invalid index: " +
text.charAt(text.length()));

 } catch (StringIndexOutOfBoundsException e) {

 System.out.println("Caught a
StringIndexOutOfBoundsException!");

 System.out.println("Exception Message: " +
e.getMessage());
 }
 }`

```

    }
}

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    System.out.print("Enter a string: ");
    String userInput = scanner.next();

    handleStringIndexOutOfBoundsException(userInput);

    scanner.close();
}
}

```

6. Write a program to demonstrate ***IllegalArgumentException***

Hint =>

- Define a variable of type String and take user input to assign a value
- Write a Method to generate the Exception. Here use the ***subString()*** and set the start index to be greater than the end index. This will generate a runtime exception and abruptly stop the program.
- Write the Method to demonstrate ***IllegalArgumentException***. Here use the ***subString()*** and set the start index to be greater than the end index. This will generate a runtime exception. Use the try-catch block to handle the ***IllegalArgumentException*** and the generic runtime exception
- From the main Firstly call the method to generate the Exception then call the method to handle the RuntimeException


```
CODE- import java.util.Scanner;

public class IllegalArgumentDemo {

    public static void generateIllegalArgumentException(String text) {

        System.out.println("Invalid substring: " + text.substring(5,
2));
    }

    public static void handleIllegalArgumentException(String text) {
        try {
            System.out.println("Invalid substring: " +
text.substring(5, 2));
        } catch (IllegalArgumentException e) {
            System.out.println("Caught IllegalArgumentException: " +
e.getMessage());
        } catch (RuntimeException e) {
            System.out.println("Caught generic RuntimeException: " +
e.getMessage());
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a string: ");
        String userInput = scanner.next();

        handleIllegalArgumentException(userInput);
    }
}
```

```

        scanner.close();
    }
}

```

7. Write a program to demonstrate **NumberFormatException**

Hint =>

- Define a variable to take user input as a String
- Use Integer.parseInt() to generate this exception. **Integer.parseInt()** is a built-in function in java.lang.Integer class to extract the number from text. In case the text does not contain numbers the method will throw NumberFormatException which is a runtime exception
- Write a Method to generate the Exception. Use **Integer.parseInt(text)** to extract number from the text. This will generate a runtime exception and abruptly stop the program.
- Write the Method to demonstrate **NumberFormatException**. Use **Integer.parseInt(text)** to extract number from the text. This will generate a runtime exception. Use the try-catch block to handle the **NumberFormatException** as well as the generic runtime exception
- From the main Firstly call the method to generate the Exception then call the method to handle the RuntimeException

CODE- `import java.util.Scanner;`

```

public class NumberFormatDemo {

```

```

    public static void generateNumberFormatException(String text) {

        int number = Integer.parseInt(text);
        System.out.println("Parsed number: " + number);
    }
}

```



```

    public static void handleNumberFormatException(String text) {
        try {
            int number = Integer.parseInt(text);
            System.out.println("Parsed number: " + number);
        } catch (NumberFormatException e) {
            System.out.println("Caught NumberFormatException: " +
e.getMessage());
        } catch (RuntimeException e) {
            System.out.println("Caught generic RuntimeException: " +
e.getMessage());
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a number (or non-numeric string
to trigger exception): ");
        String userInput = scanner.next();

        handleNumberFormatException(userInput);

        scanner.close();
    }
}

```

8. Write a program to demonstrate **ArrayIndexOutOfBoundsException**

Hint =>

- a. Define a variable of array of names and take input from the user
- b. Write a Method to generate the Exception. Here access index larger then the length of the array. This will generate a runtime exception and abruptly stop the program.

- c. Write the Method to demonstrate **ArrayIndexOutOfBoundsException**. Here access index larger then the length of the array. This will generate a runtime exception. Use the try-catch block to handle the **ArrayIndexOutOfBoundsException** and the generic runtime exception
- d. From the main Firstly call the method to generate the Exception then call the method to handle the RuntimeException

CODE- `import java.util.Scanner;`

```
public class ArrayIndexDemo {
```

```
    public static void  
generateArrayIndexOutOfBoundsException(String[] names, int index) {  
        System.out.println("Name at index " + index + ": " +  
names[index]);  
    }
```

```
    public static void  
handleArrayIndexOutOfBoundsException(String[] names, int index) {  
        try {  
            System.out.println("Name at index " + index + ": " +  
names[index]);  
        } catch (ArrayIndexOutOfBoundsException e) {  
            System.out.println("Caught ArrayIndexOutOfBoundsException:  
" + e.getMessage());  
        } catch (RuntimeException e) {  
            System.out.println("Caught generic RuntimeException: " +  
e.getMessage());  
        }  
    }
```

```
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);
```



```

        String[] names = { "Alice", "Bob", "Charlie", "Diana",
        "Ethan" };

        System.out.println("There are " + names.length + " names in
the array.");

        System.out.print("Enter an index to access: ");
        int index = scanner.nextInt();

        handleArrayIndexOutOfBoundsException(names, index);

        scanner.close();
    }
}

```

9. Write a program to convert the complete text to uppercase and compare the results

Hint =>

- Take user input using the **Scanner nextLine()** method to take the complete text into a String variable
- Write a method using the String built-in **charAt()** method to convert each character if it is lowercase to the Upper Case. Use the logic ASCII value of 'a' is 97 and 'A' is 65 so the difference is 32, similarly ASCII value of 'b' is 98 and 'B' is 66 so the difference is 32, and so on
- Write a method to compare two strings using the charAt() method and return a boolean result
- In the main() use the String built-in method **toLowerCase()** to get the Uppercase Text and compare the two strings using the user-defined method. And finally display the result

CODE- `import java.util.Scanner;`

`public class UpperCaseConverter {`

`public static String toUpperCaseManual(String text) {`

```

        StringBuilder result = new StringBuilder();

        for (int i = 0; i < text.length(); i++) {
            char ch = text.charAt(i);

            if (ch >= 'a' && ch <= 'z') {
                result.append((char) (ch - 32));
            } else {
                result.append(ch); // Keep other characters as they
are
            }
        }

        return result.toString();
    }

    public static boolean compareStrings(String str1, String str2) {
        if (str1.length() != str2.length()) {
            return false;
        }

        for (int i = 0; i < str1.length(); i++) {
            if (str1.charAt(i) != str2.charAt(i)) {
                return false;
            }
        }

        return true;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
    }

```



```

        System.out.print("Enter a line of text: ");
String input = scanner.nextLine();

String manualUpper = toUpperCaseManual(input);

String builtInUpper = input.toUpperCase();

boolean isSame = compareStrings(manualUpper,
builtInUpper);

System.out.println("\nManual Uppercase:  " + manualUpper);
System.out.println("Built-in Uppercase: " + builtInUpper);
System.out.println("Are both uppercase versions equal? " +
isSame);

    scanner.close();
}
}

```

10. Write a program to convert the complete text to lowercase and compare the results

Hint =>

- Take user input using the **Scanner nextLine()** method to take the complete text into a String variable
- Write a method using the String built-in **charAt()** method to convert each character if it is lowercase to the Upper Case. Use the logic ASCII value of 'a' is 97 and 'A' is 65 so the difference is 32, similarly ASCII value of 'b' is 98 and 'B' is 66 so the difference is 32, and so on
- Write a method to compare two strings using the charAt() method and return a boolean result
- In the main() use the String built-in method **toUpperCase()** to get the Uppercase Text and compare the two strings using the user-defined method. And finally display the result

```

CODE- import java.util.Scanner;

public class LowerCaseConverter {

    public static String toLowerCaseManual(String text) {
        StringBuilder result = new StringBuilder();

        for (int i = 0; i < text.length(); i++) {
            char ch = text.charAt(i);

            if (ch >= 'A' && ch <= 'Z') {
                result.append((char) (ch + 32));
            } else {
                result.append(ch); // Keep other characters as they
are
            }
        }

        return result.toString();
    }

    public static boolean compareStrings(String str1, String str2)
    {
        if (str1.length() != str2.length()) {
            return false;
        }

        for (int i = 0; i < str1.length(); i++) {
            if (str1.charAt(i) != str2.charAt(i)) {
                return false;
            }
        }

        return true;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a line of text: ");
        String input = scanner.nextLine();
    }
}

```



```
        String manualLower = toLowerCaseManual(input);

        String builtInLower = input.toLowerCase();

        boolean isSame = compareStrings(manualLower,
builtInLower);

        System.out.println("\nManual Lowercase:  " + manualLower);
        System.out.println("Built-in Lowercase: " + builtInLower);
        System.out.println("Are both lowercase versions equal? " +
isSame);

        scanner.close();
    }
}
```