CodInClub

Powered By BridgeLabz

Best Programming Practice

1. Use Variables including for Fixed, User Inputs, and Results
2. Use Methods instead of writing code in the main() function
3. Proper naming conventions for all variables and methods
4. Proper Program Name and Class Name
5. Handle Checked and Unchecked Exceptions wherever possible
6. Proper Method Name which indicates action taking inputs and providing result

Sample Program 1: Create a program to find all the occurrences of a character in a string using charAt() method

a. Take user input for the String and occurrences of the Character to find
b. Write a method to find all the occurrences of the characters.
    i. The logic used is to first find the number of occurrences of the character and
    ii. then create an array to store the indexes of the character
c. Call the method in the main and display the result

Java

```java
// Program to find all the occurrences of a character in a string
import java.util.Scanner;
class StringAnalyzer {
  // Method to find all the index of a character in a string using charAt()
  // method and return them in an array
  public static int[] findAllIndexes(String text, char ch) {
    // The count is used to find the number of occurrences of the character
    int count = 0;
    for (int i = 0; i < text.length(); i++) {
        if (text.charAt(i) == ch) {
            count++;
        }
    }

    // Create an array to store the indexes of the character
    int[] indexes = new int[count];
    int j = 0;
    for (int i = 0; i < text.length(); i++) {
        if (text.charAt(i) == ch) {
            indexes[j] = i;
            j++;
        }
    }
    return indexes;
```

```
  }
  public static void main(String[] args) {
    // Take user input for Text and Character to check Occurrences
    Scanner sc = new Scanner(System.in);
    System.out.print(Enter a text: ");
    String text = sc.nextLine();
    System.out.print("Enter a character to find the occurrences: ");
    char ch = sc.next().charAt(0);

    // Find the occurrences of the character
    int[] indexes = findAllIndexes(text, ch);

    // Display the occurrences of the character
    System.out.println("Indexes of the character '" + ch + "': ");
    for (int i = 0; i < indexes.length; i++) {
        System.out.print(indexes[i] + " ");
    }
  }
}
```

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

```
package Day5.LabPractice_L1;

import java.util.Scanner;

public class LP1 {

   public static boolean charAt(String stringA, String stringB){

       if (stringA.length() != stringB.length()){

           return false;

       }

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

           if (stringA.charAt(i) != stringB.charAt(i)){

               return false;

           }

       }

       return true;

   }


   public static void main(String[] args){

       Scanner input = new Scanner(System.in);

       System.out.print("Enter String A: ");

       String stringA = input.next();

       System.out.print("Enter String B: ");

       String stringB = input.next();

       boolean charAtComparision = charAt(stringA, stringB);

       boolean equalsComparision = stringA.equals(stringB);

       System.out.println("\nComparison using charAt(): " +
charAtComparision);
```

```
        System.out.println("Comparison using equals(): " +
equalsComparision);
        if(charAtComparision == equalsComparision){
            System.out.println("Both methods produce the same result.");
        }
        else{
            System.out.println("Methods produce different results.");
        }
        input.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

```
package Day5.LabPractice_L1;
import java.util.Scanner;
public class LP2 {
    public static String createSubstring(String str, int start, int end) {
        StringBuilder subStr = new StringBuilder();
        for (int i = start; i < end; i++) {
            subStr.append(str.charAt(i));
        }
        return subStr.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 the main string: ");
        String mainString = scanner.nextLine();
        System.out.print("Enter the start index: ");
        int start = scanner.nextInt();
        System.out.print("Enter the end index: ");
        int end = scanner.nextInt();
        if (start < 0 || end > mainString.length() || start >= end) {
            System.out.println("Invalid indices. Please try again.");
            return;
        }
        String manualSubstring = createSubstring(mainString, start, end);
        String builtInSubstring = mainString.substring(start, end);
        boolean isEqual = compareStrings(manualSubstring,
builtInSubstring);
        System.out.println("\nSubstring using charAt(): " +
manualSubstring);
        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 =>

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

```
package Day5.LabPractice_L1;

import java.util.Scanner;

import java.util.Arrays;

public class LP3 {

   public static char[] extractChars(String str) {

       char[] chars = new char[str.length()];

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

           chars[i] = str.charAt(i);

       }

       return chars;

   }

   public static boolean compareCharArrays(char[] arr1, char[] arr2) {

       return Arrays.equals(arr1, arr2);

   }

   public static void main(String[] args) {

       Scanner input = new Scanner(System.in);

       System.out.print("Enter a string: ");

       String inputString = input.next();

       char[] userDefinedChars = extractChars(inputString);

       char[] builtInChars = inputString.toCharArray();

       boolean areEqual = compareCharArrays(userDefinedChars,
builtInChars);

       System.out.println("Characters using user-defined method: " +
Arrays.toString(userDefinedChars));

       System.out.println("Characters using built-in toCharArray(): " +
Arrays.toString(builtInChars));
```

```
        System.out.println("Are both methods giving the same result? " +
areEqual);

        input.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

```
package Day5.LabPractice_L1;


public class LP4 {

    public static void generateException() {

        String text = null;

        System.out.println(text.length());

    }

    public static void handleException() {

        String text = null;

        try {

            System.out.println(text.length());

        } catch (NullPointerException e) {

            System.out.println("NullPointerException caught: " +
e.getMessage());

        }

    }

    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

```
package Day5.LabPractice_L1;

import java.util.Scanner;

public class LP5 {

   public static void generateException(String text) {

       System.out.println("Attempting to access an out-of-bounds
index...");

       System.out.println(text.charAt(text.length()));

   }

   public static void handleException(String text) {

       try {

           System.out.println("Attempting to access an out-of-bounds
index...");

           System.out.println(text.charAt(text.length()));

       } catch (StringIndexOutOfBoundsException e) {

           System.out.println("StringIndexOutOfBoundsException caught: "
+ e.getMessage());

       }

   }

   public static void main(String[] args) {

       Scanner input = new Scanner(System.in);

       System.out.print("Enter a string: ");

       String userInput = input.nextLine();

       handleException(userInput);
```

```
        input.close();
    }
}
```

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

   Hint =>

   a. Define a variable of type String and take user input to assign a value
   b. 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.
   c. 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
   d. From the main Firstly call the method to generate the Exception then call the method to handle the RuntimeException

```
package Day5.LabPractice_L1;

import java.util.Scanner;

public class LP6 {

    public static void generateException(String text) {

        System.out.println("Attempting to create a substring with an
invalid index range...");

        System.out.println(text.substring(5, 2));

    }

    public static void handleException(String text) {

        try {

            System.out.println("Attempting to create a substring with an
invalid index range...");

            System.out.println(text.substring(5, 2));

        } catch (IllegalArgumentException e) {

            System.out.println("IllegalArgumentException caught: " +
e.getMessage());

        } catch (RuntimeException e) {

            System.out.println("RuntimeException caught: " +
e.getMessage());

        }
```

```
    }
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String userInput = input.nextLine();
        handleException(userInput);
        input.close();
    }
}
```

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

   Hint =>

   a. Define a variable to take user input as a String
   b. 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
   c. 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.
   d. 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
   e. From the main Firstly call the method to generate the Exception then call the method to handle the RuntimeException

```
package Day5.LabPractice_L1;
import java.util.Scanner;
public class LP7 {
    public static void generateException(String text) {
        System.out.println("Attempting to parse integer from input...");
        int number = Integer.parseInt(text);
        System.out.println("Parsed Number: " + number);
    }
    public static void handleException(String text) {
```

```
        try {
            System.out.println("Attempting to parse integer from
input...");
            int number = Integer.parseInt(text);
            System.out.println("Parsed Number: " + number);
        } catch (NumberFormatException e) {
            System.out.println("NumberFormatException caught: Invalid
number format - " + e.getMessage());
        } catch (RuntimeException e) {
            System.out.println("RuntimeException caught: " +
e.getMessage());
        }
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String userInput = scanner.nextLine();
        handleException(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

```
package Day5.LabPractice_L1;

import java.util.Scanner;
```

```
public class LP8 {
    public static void generateException(String[] names, int index) {
        System.out.println("Attempting to access index: " + index);
        System.out.println("Name at index " + index + ": " +
names[index]);
    }
    public static void handleException(String[] names, int index) {
        try {
            System.out.println("Attempting to access index: " + index);
            System.out.println("Name at index " + index + ": " +
names[index]);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("ArrayIndexOutOfBoundsException caught: " +
e.getMessage());
        } catch (RuntimeException e) {
            System.out.println("RuntimeException caught: " +
e.getMessage());
        }
    }
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        String[] names = {"Alice", "Bob", "Charlie", "David", "Emma"};
        System.out.print("Enter an index to access (0 to " + (names.length
- 1) + "): ");
        int index = input.nextInt();
        handleException(names, index);
        input.close();
    }
}
```

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

   Hint =>

   a. Take user input using the ***Scanner nextLine()*** method to take the complete text into a String variable

b. Write a method using the String built-in ***charAt()*** method to convert each character if it is lowercase to the uppercase. 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
c. Write a method to compare two strings using the charAt() method and return a boolean result
d. 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

```
package Day5.LabPractice_L1;

import java.util.Scanner;

public class LP9 {

   public static String convertToUpperCase(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);

           }

       }

       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 input = new Scanner(System.in);

        System.out.print("Enter a text: ");

        String inputText = input.nextLine();

        String builtInUpperCase = inputText.toUpperCase();

        String customUpperCase = convertToUpperCase(inputText);

        boolean areEqual = compareStrings(builtInUpperCase,
customUpperCase);

        System.out.println("\nOriginal Text: " + inputText);

        System.out.println("Custom Uppercase Conversion: " +
customUpperCase);

        System.out.println("Built-in Uppercase Conversion: " +
builtInUpperCase);

        System.out.println("Are both conversions equal? " + areEqual);

        input.close();

    }

}
```

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

    Hint =>

    a. Take user input using the ***Scanner*** ***nextLine()*** method to take the complete text into a String variable
    b. Write a method using the String built-in ***charAt()*** method to convert each character if it is uppercase to the lowercase. 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
    c. Write a method to compare two strings using the charAt() method and return a boolean result
    d. In the main() use the String built-in method ***toLowerCase()*** to get the lowercase text and compare the two strings using the user-defined method. And finally display the result

```
package Day5.LabPractice_L1;

import java.util.Scanner;

public class LP10 {

    public static String convertToLowerCase(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);
            }
        }
        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 input = new Scanner(System.in);
        System.out.print("Enter a text: ");
        String inputText = input.nextLine();
        String builtInLowerCase = inputText.toLowerCase();
        String customLowerCase = convertToLowerCase(inputText);
        boolean areEqual = compareStrings(builtInLowerCase,
customLowerCase);
        System.out.println("\nOriginal Text: " + inputText);
        System.out.println("Custom Lowercase Conversion: " +
customLowerCase);
        System.out.println("Built-in Lowercase Conversion: " +
builtInLowerCase);
        System.out.println("Are both conversions equal? " + areEqual);
```

```
        input.close();
    }
}
```