

Level 1 String

```
import java.util.*;
```

1. Compare Strings

```
public static void compareStringsDemo() {  
    Scanner sc = new Scanner(System.in);  
    System.out.print("Enter first string: ");  
    String s1 = sc.next();  
    System.out.print("Enter second string: ");  
    String s2 = sc.next();  
  
    boolean manual = compareStrings(s1, s2);  
    boolean builtin = s1.equals(s2);  
  
    System.out.println("Manual comparison: " + manual);  
    System.out.println("Built-in equals(): " + builtin);  
}
```

2. Substring using charAt() vs substring()

```
public static String customSubstring(String text, int start, int end) {  
    String result = "";  
    for (int i = start; i < end && i < text.length(); i++) {  
        result += text.charAt(i);  
    }  
    return result;  
}
```

```
public static void substringDemo() {  
    Scanner sc = new Scanner(System.in);
```

```

System.out.print("Enter text: ");
String text = sc.next();
System.out.print("Enter start index: ");
int start = sc.nextInt();
System.out.print("Enter end index: ");
int end = sc.nextInt();

String manual = customSubstring(text, start, end);
String builtin = text.substring(start, end);

System.out.println("Manual substring: " + manual);
System.out.println("Built-in substring: " + builtin);
System.out.println("Are they equal? " + compareStrings(manual, builtin));
}

```

3. Convert to Char Array

```

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

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

```

```
        return true;
    }

    public static void charArrayDemo() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter text: ");
        String text = sc.next();

        char[] manual = customToCharArray(text);
        char[] builtin = text.toCharArray();

        System.out.println("Arrays match? " + compareCharArrays(manual, builtin));
    }
```

4. NullPointerException Demo

```
    public static void generateNPE() {
        String text = null;
        System.out.println(text.length()); // will throw NPE
    }

    public static void handleNPE() {
        try {
            String text = null;
            System.out.println(text.length());
        } catch (NullPointerException e) {
            System.out.println("Handled NullPointerException: " + e);
        }
    }
}
```

5. StringIndexOutOfBoundsException Demo

```
public static void generateSIOOBE() {  
    Scanner sc = new Scanner(System.in);  
    System.out.print("Enter text: ");  
    String text = sc.next();  
    System.out.println(text.charAt(text.length())); // out of bounds  
}
```

```
public static void handleSIOOBE() {  
    try {  
        Scanner sc = new Scanner(System.in);  
        System.out.print("Enter text: ");  
        String text = sc.next();  
        System.out.println(text.charAt(text.length()));  
    } catch (StringIndexOutOfBoundsException e) {  
        System.out.println("Handled StringIndexOutOfBoundsException: " + e);  
    }  
}
```

6. IllegalArgumentException Demo

```
public static void generateIAE() {  
    Scanner sc = new Scanner(System.in);  
    System.out.print("Enter text: ");  
    String text = sc.next();  
    System.out.println(text.substring(5, 2)); // invalid range  
}
```

```
public static void handleIAE() {  
    try {
```

```

Scanner sc = new Scanner(System.in);

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

String text = sc.next();

System.out.println(text.substring(5, 2));
} catch (IllegalArgumentException e) {

    System.out.println("Handled IllegalArgumentException: " + e);
} catch (RuntimeException e) {

    System.out.println("Handled RuntimeException: " + e);
}
}

```

7. NumberFormatException Demo

```

public static void generateNFE() {

    Scanner sc = new Scanner(System.in);

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

    String text = sc.next();

    int num = Integer.parseInt(text); // throws NFE if not number

    System.out.println("Parsed number: " + num);

}

public static void handleNFE() {

    try {

        Scanner sc = new Scanner(System.in);

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

        String text = sc.next();

        int num = Integer.parseInt(text);

        System.out.println("Parsed number: " + num);

    } catch (NumberFormatException e) {

        System.out.println("Handled NumberFormatException: " + e);
    }
}

```

```

    } catch (RuntimeException e) {
        System.out.println("Handled RuntimeException: " + e);
    }
}

```

8. ArrayIndexOutOfBoundsException Demo

```

public static void generateAIOOBE() {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter size of array: ");
    int n = sc.nextInt();
    String[] arr = new String[n];
    System.out.println(arr[n]); // invalid access
}

```

```

public static void handleAIOOBE() {
    try {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter size of array: ");
        int n = sc.nextInt();
        String[] arr = new String[n];
        System.out.println(arr[n]);
    } catch (ArrayIndexOutOfBoundsException e) {
        System.out.println("Handled ArrayIndexOutOfBoundsException: " + e);
    } catch (RuntimeException e) {
        System.out.println("Handled RuntimeException: " + e);
    }
}

```

9. Convert to Uppercase manually vs built-in

```
public static String customToUpper(String text) {  
    String result = "";  
    for (int i = 0; i < text.length(); i++) {  
        char c = text.charAt(i);  
        if (c >= 'a' && c <= 'z') {  
            result += (char)(c - 32);  
        } else {  
            result += c;  
        }  
    }  
    return result;  
}
```

```
public static void uppercaseDemo() {  
    Scanner sc = new Scanner(System.in);  
    System.out.print("Enter text: ");  
    String text = sc.nextLine();  
  
    String manual = customToUpper(text);  
    String builtin = text.toUpperCase();  
  
    System.out.println("Manual uppercase: " + manual);  
    System.out.println("Built-in uppercase: " + builtin);  
    System.out.println("Match? " + compareStrings(manual, builtin));  
}
```

// 10. Convert to Lowercase manually vs built-in

```
public static String customToLower(String text) {
```

```

String result = "";
for (int i = 0; i < text.length(); i++) {
    char c = text.charAt(i);
    if (c >= 'A' && c <= 'Z') {
        result += (char)(c + 32);
    } else {
        result += c;
    }
}
return result;
}

```

```

public static void lowercaseDemo() {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter text: ");
    String text = sc.nextLine();

    String manual = customToLower(text);
    String builtin = text.toLowerCase();

    System.out.println("Manual lowercase: " + manual);
    System.out.println("Built-in lowercase: " + builtin);
    System.out.println("Match? " + compareStrings(manual, builtin));
}
}

```