--- Question1_CompareStrings.java ---

```
import java.util.Scanner;

public class Question1_CompareStrings {

    public static boolean compareUsingCharAt(String s1, String s2) {
        if (s1.length() != s2.length()) {
            return false;
        }
        for (int i = 0; i < s1.length(); i++) {
            if (s1.charAt(i) != s2.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 str1 = scanner.next();

        System.out.print("Enter second string: ");
        String str2 = scanner.next();

        boolean resultCharAt = compareUsingCharAt(str1, str2);
        boolean resultEquals = str1.equals(str2);

        System.out.println("Comparison using charAt(): " + resultCharAt);
        System.out.println("Comparison using equals(): " + resultEquals);

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

        scanner.close();
    }
}
```

--- Question2_SubstringComparison.java ---

```
import java.util.Scanner;
```

```
public class Question2_SubstringComparison {

    public static String substringUsingCharAt(String str, int start, int end) {
        StringBuilder sb = new StringBuilder();
        for (int i = start; i < end; i++) {
            sb.append(str.charAt(i));
        }
        return sb.toString();
    }

    public static boolean compareUsingCharAt(String s1, String s2) {
        if (s1.length() != s2.length()) {
            return false;
        }
        for (int i = 0; i < s1.length(); i++) {
            if (s1.charAt(i) != s2.charAt(i)) {
                return false;
            }
        }
        return true;
    }

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

        System.out.print("Enter the 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 substringCharAt = substringUsingCharAt(input, start, end);
        String substringBuiltIn = input.substring(start, end);

        System.out.println("Substring using charAt(): " + substringCharAt);
        System.out.println("Substring using substring(): " + substringBuiltIn);

        boolean comparisonResult = compareUsingCharAt(substringCharAt, substringBuiltIn);
        System.out.println("Are both substrings equal? " + comparisonResult);
```

```
        scanner.close();
    }
}
```

--- Question3_CharArrayComparison.java ---

```
import java.util.Arrays;
import java.util.Scanner;

public class Question3_CharArrayComparison {

    public static char[] getChars(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) {
        if (arr1.length != arr2.length) {
            return false;
        }
        for (int i = 0; i < arr1.length; i++) {
            if (arr1[i] != arr2[i]) {
                return false;
            }
        }
        return true;
    }

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

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

        char[] userDefinedChars = getChars(input);
        char[] builtInChars = input.toCharArray();

        System.out.println("User-defined char array: " + Arrays.toString(userDefinedChars));
        System.out.println("Built-in char array: " + Arrays.toString(builtInChars));

        boolean areEqual = compareCharArrays(userDefinedChars, builtInChars);
        System.out.println("Are both char arrays equal? " + areEqual);
```

```
        scanner.close();
    }
}
```

--- Question4_NullPointerExceptionDemo.java ---

```
public class Question4_NullPointerExceptionDemo {

    public static void generateNullPointerException() {
        String text = null;
        // This will throw NullPointerException
        System.out.println(text.length());
    }

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

    public static void main(String[] args) {
        // Call method to generate exception (will cause program to stop if uncommented)
        // generateNullPointerException();

        // Call method to handle exception
        handleNullPointerException();
    }
}
```

--- Question5_StringIndexOutOfBoundsExceptionDemo.java ---

```
import java.util.Scanner;

public class Question5_StringIndexOutOfBoundsExceptionDemo {

    public static void generateException(String str) {
        // This will throw StringIndexOutOfBoundsException if index is out of range
        System.out.println(str.charAt(str.length()));
    }

    public static void handleException(String str) {
        try {
```

```
            System.out.println(str.charAt(str.length()));
        } catch (StringIndexOutOfBoundsException e) {
            System.out.println("Caught StringIndexOutOfBoundsException: " + e.getMessage());
        }
    }

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

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

        // Uncomment to generate exception (program will stop)
        // generateException(input);

        // Handle exception gracefully
        handleException(input);

        scanner.close();
    }
}
```

--- Question6_IllegalArgumentExceptionDemo.java ---

```
import java.util.Scanner;

public class Question6_IllegalArgumentExceptionDemo {

    public static void generateException(String str) {
        // This will throw IllegalArgumentException if start > end
        System.out.println(str.substring(5, 2));
    }

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

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

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

        // Uncomment to generate exception (program will stop)
        // generateException(input);

        // Handle exception gracefully
        handleException(input);

        scanner.close();
    }
}
```

--- Question7_NumberFormatExceptionDemo.java ---

```
import java.util.Scanner;

public class Question7_NumberFormatExceptionDemo {

    public static void generateException(String text) {
        // This will throw NumberFormatException if text is not a valid number
        int number = Integer.parseInt(text);
        System.out.println("Parsed number: " + number);
    }

    public static void handleException(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 RuntimeException: " + e.getMessage());
        }
    }

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

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

        // Uncomment to generate exception (program will stop)
        // generateException(input);
```

```
        // Handle exception gracefully
        handleException(input);

        scanner.close();
    }
}
```

--- Question8_ArrayIndexOutOfBoundsExceptionDemo.java ---

```
import java.util.Scanner;

public class Question8_ArrayIndexOutOfBoundsExceptionDemo {

    public static void generateException(String[] names) {
        // This will throw ArrayIndexOutOfBoundsException if index is out of range
        System.out.println(names[names.length]);
    }

    public static void handleException(String[] names) {
        try {
            System.out.println(names[names.length]);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Caught ArrayIndexOutOfBoundsException: " + e.getMessage());
        } catch (RuntimeException e) {
            System.out.println("Caught RuntimeException: " + e.getMessage());
        }
    }

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

        System.out.print("Enter number of names: ");
        int n = scanner.nextInt();
        scanner.nextLine(); // consume newline

        String[] names = new String[n];
        for (int i = 0; i < n; i++) {
            System.out.print("Enter name " + (i + 1) + ": ");
            names[i] = scanner.nextLine();
        }

        // Uncomment to generate exception (program will stop)
        // generateException(names);
```

```
        // Handle exception gracefully
        handleException(names);

        scanner.close();
    }
}
```

--- Question9_UppercaseConversion.java ---

```
import java.util.Scanner;

public class Question9_UppercaseConversion {

    public static String toUpperCaseUsingCharAt(String str) {
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < str.length(); i++) {
            char ch = str.charAt(i);
            if (ch >= 'a' && ch <= 'z') {
                sb.append((char)(ch - 32));
            } else {
                sb.append(ch);
            }
        }
        return sb.toString();
    }

    public static boolean compareUsingCharAt(String s1, String s2) {
        if (s1.length() != s2.length()) {
            return false;
        }
        for (int i = 0; i < s1.length(); i++) {
            if (s1.charAt(i) != s2.charAt(i)) {
                return false;
            }
        }
        return true;
    }

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

        System.out.print("Enter the complete text: ");
        String input = scanner.nextLine();

        String upperCharAt = toUpperCaseUsingCharAt(input);
```

```
        String upperBuiltIn = input.toUpperCase();

        System.out.println("Uppercase using charAt(): " + upperCharAt);
        System.out.println("Uppercase using toUpperCase(): " + upperBuiltIn);

        boolean isEqual = compareUsingCharAt(upperCharAt, upperBuiltIn);
        System.out.println("Are both uppercase strings equal? " + isEqual);

        scanner.close();
    }
}
```

--- Question10_LowercaseConversion.java ---

```
import java.util.Scanner;

public class Question10_LowercaseConversion {

    public static String toLowerCaseUsingCharAt(String str) {
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < str.length(); i++) {
            char ch = str.charAt(i);
            if (ch >= 'A' && ch <= 'Z') {
                sb.append((char)(ch + 32));
            } else {
                sb.append(ch);
            }
        }
        return sb.toString();
    }

    public static boolean compareUsingCharAt(String s1, String s2) {
        if (s1.length() != s2.length()) {
            return false;
        }
        for (int i = 0; i < s1.length(); i++) {
            if (s1.charAt(i) != s2.charAt(i)) {
                return false;
            }
        }
        return true;
    }

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

```
        System.out.print("Enter the complete text: ");
        String input = scanner.nextLine();

        String lowerCharAt = toLowerCaseUsingCharAt(input);
        String lowerBuiltIn = input.toLowerCase();

        System.out.println("Lowercase using charAt(): " + lowerCharAt);
        System.out.println("Lowercase using toLowerCase(): " + lowerBuiltIn);

        boolean isEqual = compareUsingCharAt(lowerCharAt, lowerBuiltIn);
        System.out.println("Are both lowercase strings equal? " + isEqual);

        scanner.close();
    }
}
```