**RegNo:** 23MCA1030 **Name**: Vinayak Kumar Singh

Java Programming Lab (PMCA502P)

**Exercise 7b:** Index Out of Bounds Exception Handling

**1:** Write a Java program that initializes an array of integers with some values. Prompt the user to enter an index and then display the value at that index. If the user enters an index that is outside the bounds of the array, catch the IndexOutOfBoundsException, display an appropriate error message, and prompt the user to re-enter the index until a valid index is provided.

Your program should:

Initialize an array of integers with some values (you can define these values in the code).

Prompt the user to enter an index.

Display the value at that index.

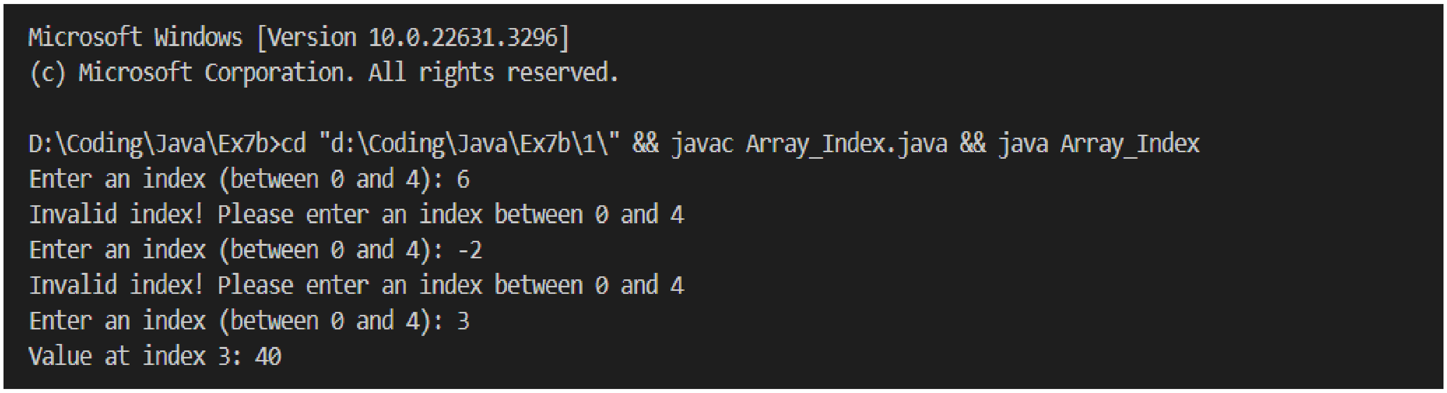
If the user enters an invalid index (negative or greater than the size of the array), catch the IndexOutOfBoundsException, display an error message, and prompt the user to re-enter the index until a valid index is provided.

Finally, print a message with the value at the valid index entered by the user.

**Code:**

```
import java.util.Scanner;
public class Array_Index {
    public static void main(String[] args) {
        // Initialize an array of integers
        int[] numbers = {10, 20, 30, 40, 50};
        Scanner scanner = new Scanner(System.in);
        int index;
        do {
            System.out.print("Enter an index (between 0 and " + (numbers.length - 1) + "): ");
            index = scanner.nextInt();
            try {
                // Display the value when entered index
                System.out.println("Value at index " + index + ": " + numbers[index]);
                break; // Exit the loop whenvalid index entered
            } catch (IndexOutOfBoundsException e) {
                // Catch the IndexOutOfBoundsException and display an error message
                System.out.println("Invalid index kindly enter an index between 0 and " +
(numbers.length - 1));
            }
        } while (true); // Loop until valid index is provided
        scanner.close();
    }
}
```

**Output:**



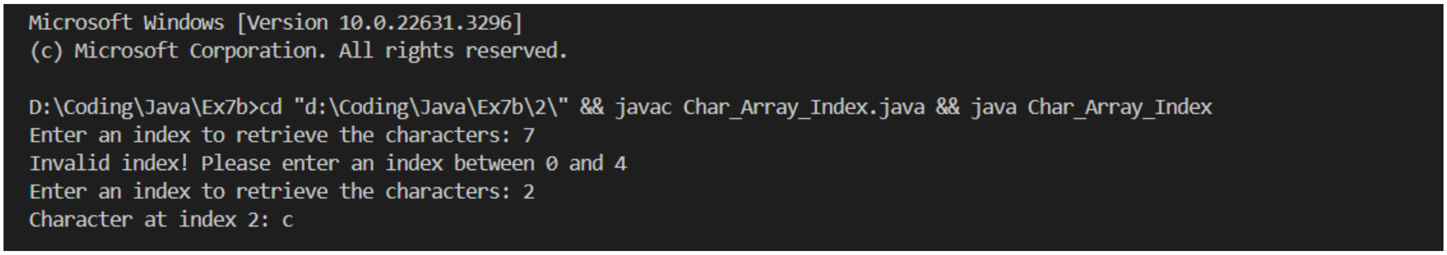
```
Microsoft Windows [Version 10.0.22631.3296]
(c) Microsoft Corporation. All rights reserved.

D:\Coding\Java\Ex7b>cd "d:\Coding\Java\Ex7b\1\" && javac Array_Index.java && java Array_Index
Enter an index (between 0 and 4): 6
Invalid index! Please enter an index between 0 and 4
Enter an index (between 0 and 4): -2
Invalid index! Please enter an index between 0 and 4
Enter an index (between 0 and 4): 3
Value at index 3: 40
```

2: Write a Java program that initializes an array of characters with some values. Prompt the user to enter an index and display the character at that index. If the user enters an index that is outside the bounds of the array, catch the IndexOutOfBoundsException, display an appropriate error message, and prompt the user to re-enter the index until a valid index is provided.

**Code:**

```
import java.util.Scanner;
public class Char_Array_Index {
    public static void main(String[] args) {
        // Initialize an array of characters
        char[] characters = {'a', 'b', 'c', 'd', 'e'};
        Scanner scanner = new Scanner(System.in);
        int index;
        do {
            System.out.print("Enter an index to retrieve the characters: ");
            index = scanner.nextInt();
            try {
                // Display character at  entered index
                System.out.println("Character at index " + index + ": " + characters[index]);
                break; // Exit the loop if valid index entered
            } catch (IndexOutOfBoundsException e) {
                // Catch the IndexOutOfBoundsException & display error message
                System.out.println("Invalid index! Please enter an index between 0 and " +
(characters.length - 1));
            }
        } while (true); // Loop until valid index is entered
        scanner.close();
    }
}
```

**Output:**



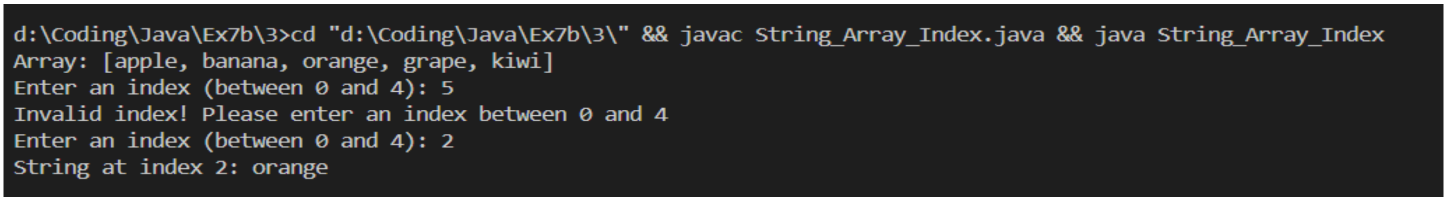
```
Microsoft Windows [Version 10.0.22631.3296]
(c) Microsoft Corporation. All rights reserved.

D:\Coding\Java\Ex7b>cd "d:\Coding\Java\Ex7b\2\" && javac Char_Array_Index.java && java Char_Array_Index
Enter an index to retrieve the characters: 7
Invalid index! Please enter an index between 0 and 4
Enter an index to retrieve the characters: 2
Character at index 2: c
```

3. Write a Java program that initializes an array of strings with some values. Prompt the user to enter an index and display the string at that index. If the user enters an index that is outside the bounds of the array, catch the IndexOutOfBoundsException, display an appropriate error message, and prompt the user to re-enter the index until a valid index is provided.

**Code:**

```
import java.util.Scanner;
import java.util.Arrays;
public class String_Array_Index{
    public static void main(String[] args) {
        // Initialize an array of strings
        String[] strings = {"apple", "banana", "orange", "grape", "kiwi"};
        System.out.println("Array: " + Arrays.toString(strings));
        Scanner scanner = new Scanner(System.in);
        int index;
        do {
            System.out.print("Enter an index (between 0 and " + (strings.length - 1) + "): ");
            index = scanner.nextInt();
            try {
                // Display the string when entered index
                System.out.println("String at index " + index + ": " + strings[index]);
                break; // Exit the loop if enterd index is valid
            } catch (IndexOutOfBoundsException e) {
                // Catch IndexOutOfBoundsException and display error message
                System.out.println("Invalid index! Please enter an index between 0 and " +
(strings.length - 1));
            }
        } while (true); // Loop until a valid index is provided
        scanner.close();
    }
}
```

**Output:**



```
d:\Coding\Java\Ex7b\3>cd "d:\Coding\Java\Ex7b\3\" && javac String_Array_Index.java && java String_Array_Index
Array: [apple, banana, orange, grape, kiwi]
Enter an index (between 0 and 4): 5
Invalid index! Please enter an index between 0 and 4
Enter an index (between 0 and 4): 2
String at index 2: orange
```

4. Write a Java program that takes a sentence as input from the user. Prompt the user to enter a start index and an end index to extract a substring from the sentence. If the start index or end index is invalid (negative or greater than the length of the sentence), catch the StringIndexOutOfBoundsException, display an appropriate error message, and prompt the user to re-enter the indices until valid indices are provided. Finally, print the extracted substring.

Your program should:

Prompt the user to enter a sentence.

Prompt the user to enter a start index.

Prompt the user to enter an end index.

Extract the substring from the sentence using the provided indices.

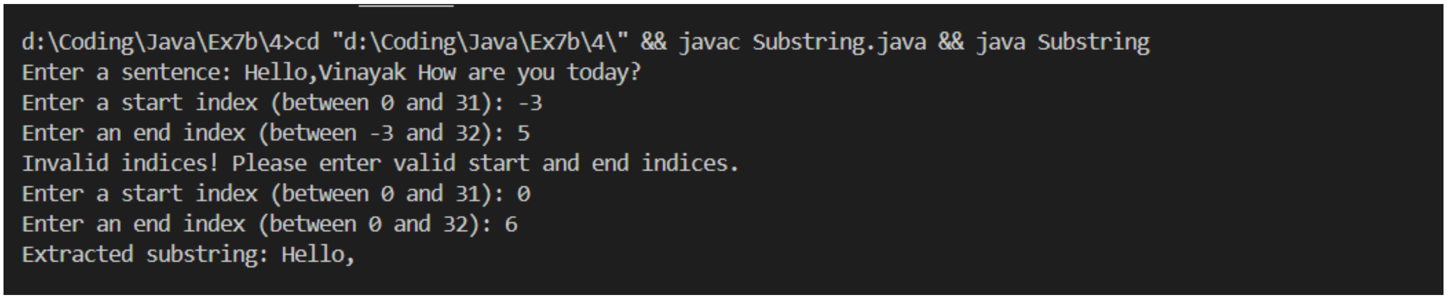
If the user enters invalid indices, catch the StringIndexOutOfBoundsException, display an error message, and prompt the user to re-enter the indices until valid indices are provided.

Print the extracted substring.

**Code:**

```
import java.util.Scanner;
public class Substring {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a sentence: ");
        String sentence = scanner.nextLine();
        int startIndex, endIndex;
        String substring;
        do {
            // Prompt to enter a start index
            System.out.print("Enter a start index (between 0 and " + (sentence.length() - 1) + "): ");
            startIndex = scanner.nextInt();
            // Prompt  to enter an end index
            System.out.print("Enter an end index (between " + startIndex + " and " +
(sentence.length()) + "): ");
            endIndex = scanner.nextInt();
            scanner.nextLine();
            try {
                // Extract substring from sentence using provided indices
                substring = sentence.substring(startIndex, endIndex);
                System.out.println("Extracted substring: " + substring);
                break; // Exit loop if valid indices are provided
            } catch (StringIndexOutOfBoundsException e) {
                // Catch StringIndexOutOfBoundsException & display  error message
                System.out.println("Invalid indices! Please enter valid start and end indices.");
            }
        } while (true); // Loop until valid indices are provided
        scanner.close();
    }
}
```

**Output:**



```
d:\Coding\Java\Ex7b\4>cd "d:\Coding\Java\Ex7b\4\" && javac Substring.java && java Substring
Enter a sentence: Hello,Vinayak How are you today?
Enter a start index (between 0 and 31): -3
Enter an end index (between -3 and 32): 5
Invalid indices! Please enter valid start and end indices.
Enter a start index (between 0 and 31): 0
Enter an end index (between 0 and 32): 6
Extracted substring: Hello,
```

5. Write a Java program that simulates a simple login system. The program should prompt the user to enter a username and password. You will have a predefined set of valid usernames and passwords stored in an array or another appropriate data structure. If the entered username or password is null, catch the NullPointerException, display an appropriate error message, and prompt the user to re-enter the username or password until a non-null value is provided. If the entered username and password combination is invalid (not found in the predefined set), display an error message accordingly. If the combination is valid, display a success message.

Your program should:

Define a set of valid usernames and passwords.

Prompt the user to enter a username.

Prompt the user to enter a password.

Check for NullPointerException when validating the username and password.

Check if the entered username and password combination is valid.

If the combination is valid, print a success message. If not, display an appropriate error message and prompt the user to try again.

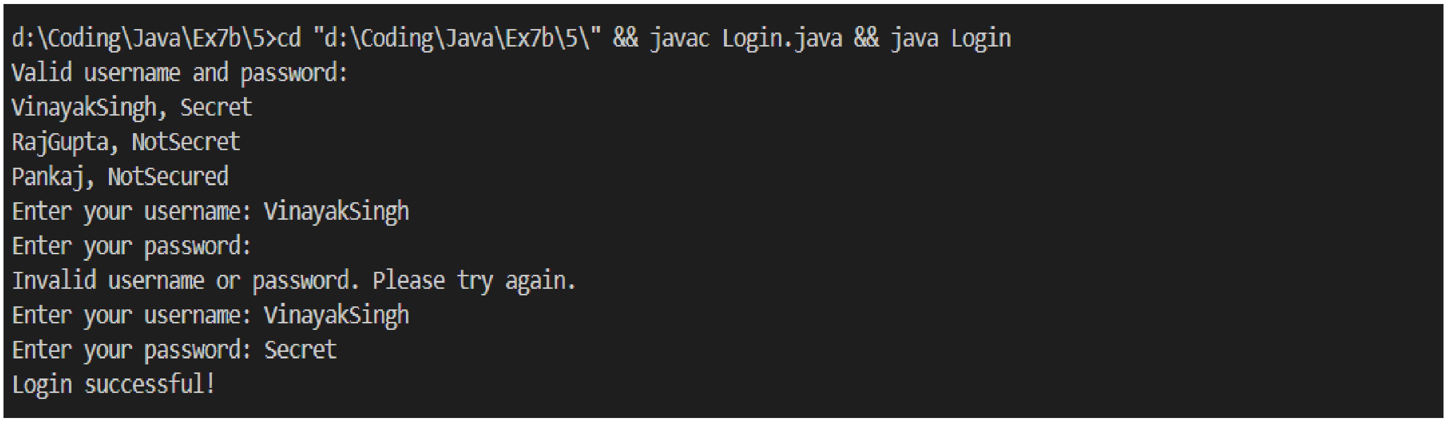
Repeat steps 2-6 until the user enters a valid username and password combination.

**Code:**

```
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;
public class Login {
    private static final Map<String, String> VALID_CREDENTIALS = new HashMap<>();
    static {
        // Define a set of valid usernames and passwords
        VALID_CREDENTIALS.put("VinayakSingh", "Secret");
        VALID_CREDENTIALS.put("RajGupta", "NotSecret");
        VALID_CREDENTIALS.put("Pankaj", "NotSecured");
    }

    public static void main(String[] args) {
        System.out.println("Valid username and password:");
        for (Map.Entry<String, String> entry : VALID_CREDENTIALS.entrySet()) {
            System.out.println(entry.getKey() + ", " + entry.getValue());
        }
        Scanner scanner = new Scanner(System.in);
        String username, password;
        do {
            System.out.print("Enter your username: ");
            username = scanner.nextLine();
            System.out.print("Enter your password: ");
            password = scanner.nextLine();
            try {
                // Check if entered username and password combination is valid
                if (VALID_CREDENTIALS.containsKey(username) &&
VALID_CREDENTIALS.get(username).equals(password)) {
```

```
                    System.out.println("Login successful!");
                    break; // Exit loop if valid credentials are provided
                } else {
                    System.out.println("Invalid username or password. Please try again.");
                }
            } catch (NullPointerException e) {
                // Catch NullPointerException and display error message
                System.out.println("Null values are not allowed for username or password. Please try again.");
            }
        } while (true); // Loop until valid credentials are provided
        scanner.close();
    }
}
```

**Output:**



```
d:\Coding\Java\Ex7b\5>cd "d:\Coding\Java\Ex7b\5\" && javac Login.java && java Login
Valid username and password:
VinayakSingh, Secret
RajGupta, NotSecret
Pankaj, NotSecured
Enter your username: VinayakSingh
Enter your password:
Invalid username or password. Please try again.
Enter your username: VinayakSingh
Enter your password: Secret
Login successful!
```