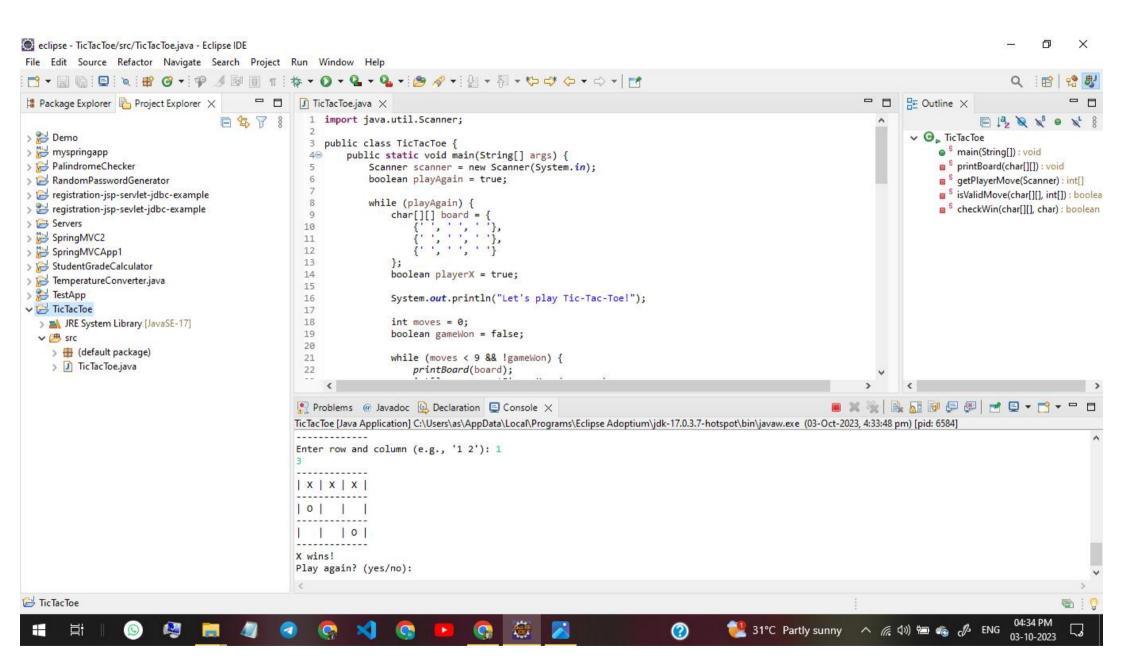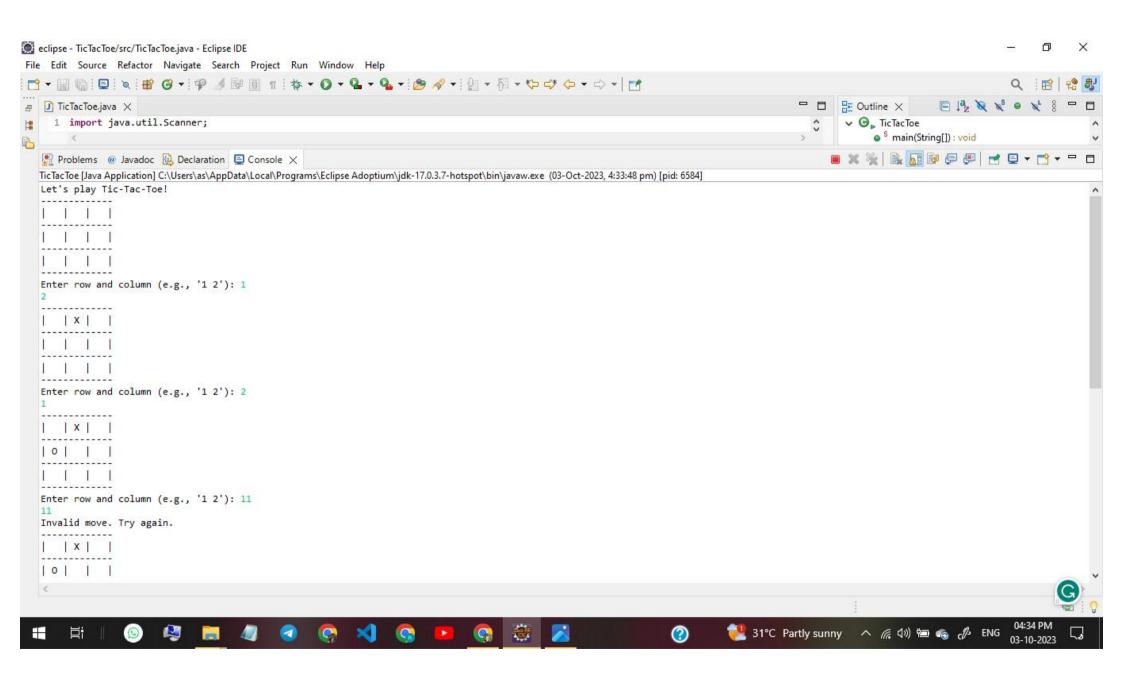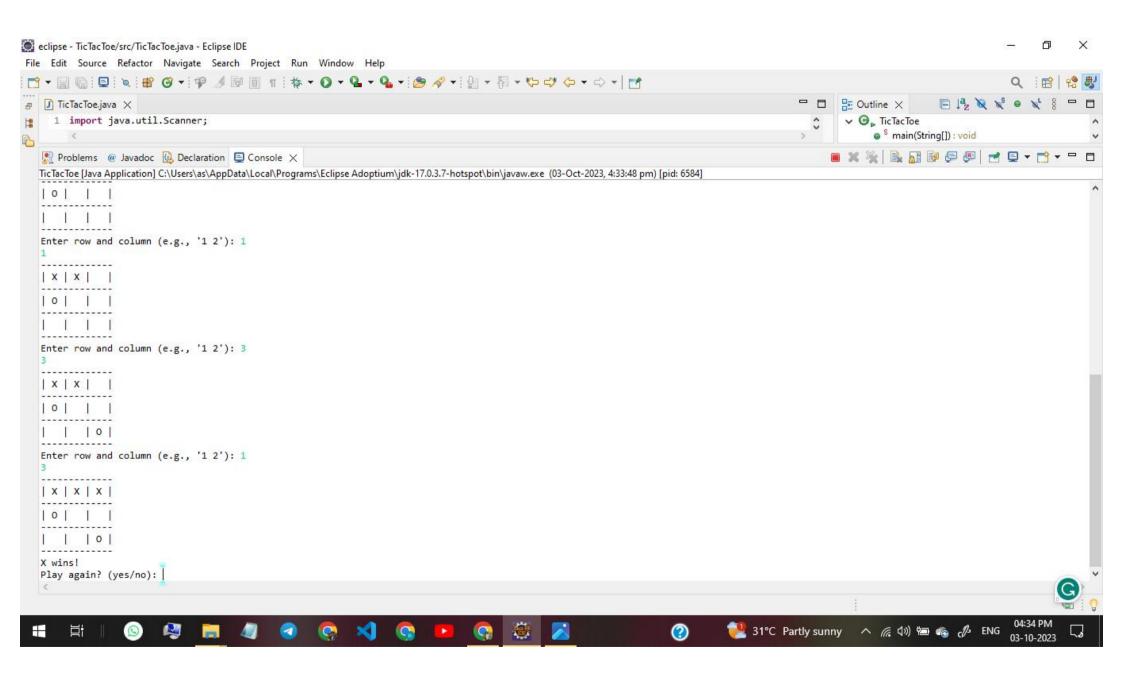# Level 2

Task 1

Task: Tic-Tac-Toe Game

Description: Implement a two-player tic-tac-toe game. Display the game board and prompt each player to enter their moves. Check for a winning condition or a draw after each move, and display the result accordingly. Allow the players to play multiple rounds if desired.

Skills: Arrays or matrices, loops, conditional statements.

File   Edit   Source   Refactor   Navigate   Search   Project   Run   Window   Help

Package Explorer   Project Explorer ×

- Demo
- myspringapp
- PalindromeChecker
- RandomPasswordGenerator
- registration-jsp-servlet-jdbc-example
- registration-jsp-sevlet-jdbc-example
- Servers
- SpringMVC2
- SpringMVCApp1
- StudentGradeCalculator
- TemperatureConverter.java
- TestApp
- TicTacToe
  - JRE System Library [JavaSE-17]
  - src
    - (default package)
    - TicTacToe.java

TicTacToe.java ×

```java
1  import java.util.Scanner;
2
3  public class TicTacToe {
4      public static void main(String[] args) {
5          Scanner scanner = new Scanner(System.in);
6          boolean playAgain = true;
7
8          while (playAgain) {
9              char[][] board = {
10                 {' ', ' ', ' '},
11                 {' ', ' ', ' '},
12                 {' ', ' ', ' '}
13             };
14             boolean playerX = true;
15
16             System.out.println("Let's play Tic-Tac-Toe!");
17
18             int moves = 0;
19             boolean gameWon = false;
20
21             while (moves < 9 && !gameWon) {
22                 printBoard(board);
```

Outline ×

- TicTacToe
  - main(String[]) : void
  - printBoard(char[][]) : void
  - getPlayerMove(Scanner) : int[]
  - isValidMove(char[][], int[]) : boolea
  - checkWin(char[][], char) : boolean

Problems   @ Javadoc   Declaration   Console ×

TicTacToe [Java Application] C:\Users\as\AppData\Local\Programs\Eclipse Adoptium\jdk-17.0.3.7-hotspot\bin\javaw.exe (03-Oct-2023, 4:33:48 pm) [pid: 6584]

```
-------------
Enter row and column (e.g., '1 2'): 1
3
-------------
| X | X | X |
-------------
| O |   |   |
-------------
|   |   | O |
-------------
X wins!
Play again? (yes/no):
```

TicTacToe

31°C   Partly sunny   ENG   04:34 PM
                                          03-10-2023

TicTacToe.java ✕   Outline ✕

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

TicTacToe
   main(String[]) : void

Problems   @ Javadoc   Declaration   Console ✕

```
Let's play Tic-Tac-Toe!
-------------
|   |   |   |
-------------
|   |   |   |
-------------
|   |   |   |
-------------
Enter row and column (e.g., '1 2'): 1
2
-------------
|   | X |   |
-------------
|   |   |   |
-------------
|   |   |   |
-------------
Enter row and column (e.g., '1 2'): 2
1
-------------
|   | X |   |
-------------
| O |   |   |
-------------
|   |   |   |
-------------
Enter row and column (e.g., '1 2'): 11
11
Invalid move. Try again.
-------------
|   | X |   |
-------------
| O |   |   |
```

31°C Partly sunny   ENG   04:34 PM 03-10-2023

File   Edit   Source   Refactor   Navigate   Search   Project   Run   Window   Help

TicTacToe.java ×

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

Outline ×

TicTacToe
  main(String[]) : void

Problems   @ Javadoc   Declaration   Console ×

TicTacToe [Java Application] C:\Users\as\AppData\Local\Programs\Eclipse Adoptium\jdk-17.0.3.7-hotspot\bin\javaw.exe (03-Oct-2023, 4:33:48 pm) [pid: 6584]

```
| O |   |   |
------------
|   |   |   |
------------
Enter row and column (e.g., '1 2'): 1
1
------------
| X | X |   |
------------
| O |   |   |
------------
|   |   |   |
------------
Enter row and column (e.g., '1 2'): 3
3
------------
| X | X |   |
------------
| O |   |   |
------------
|   |   | O |
------------
Enter row and column (e.g., '1 2'): 1
3
------------
| X | X | X |
------------
| O |   |   |
------------
|   |   | O |
------------
X wins!
Play again? (yes/no): |
```

31°C Partly sunny   ENG   04:34 PM
                                03-10-2023

Level 2

Task 2

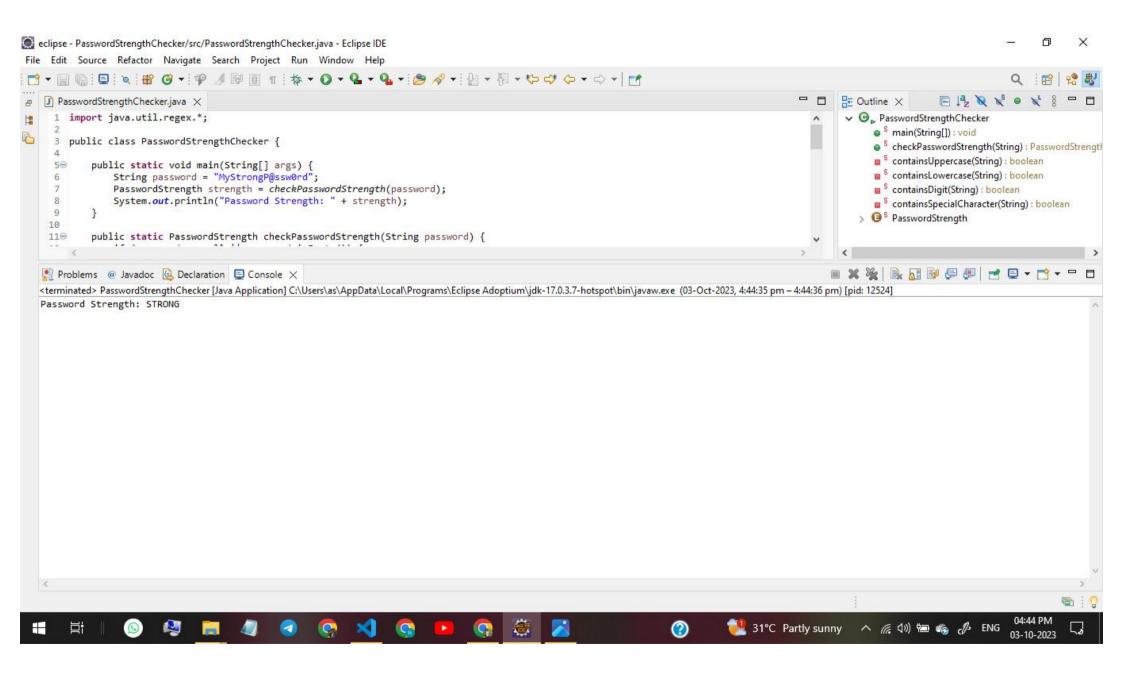**Task: Password Strength Checker**

Description: Create a program that checks the strength of a password. Prompt the user to input a password and analyze its strength based on certain criteria, such as length, presence of uppercase letters, lowercase letters, numbers, and special characters. Provide feedback on the password strength.

Skills: String   manipulation,   conditional statements.

File   Edit   Source   Refactor   Navigate   Search   Project   Run   Window   Help

PasswordStrengthChecker.java ×

```java
1  import java.util.regex.*;
2
3  public class PasswordStrengthChecker {
4
5      public static void main(String[] args) {
6          String password = "MyStrongP@ssw0rd";
7          PasswordStrength strength = checkPasswordStrength(password);
8          System.out.println("Password Strength: " + strength);
9      }
10
11     public static PasswordStrength checkPasswordStrength(String password) {
```

Outline ×

- PasswordStrengthChecker
  - main(String[]) : void
  - checkPasswordStrength(String) : PasswordStrength
  - containsUppercase(String) : boolean
  - containsLowercase(String) : boolean
  - containsDigit(String) : boolean
  - containsSpecialCharacter(String) : boolean
  - PasswordStrength

Problems   @ Javadoc   Declaration   Console ×

<terminated> PasswordStrengthChecker [Java Application] C:\Users\as\AppData\Local\Programs\Eclipse Adoptium\jdk-17.0.3.7-hotspot\bin\javaw.exe (03-Oct-2023, 4:44:35 pm – 4:44:36 pm) [pid: 12524]

Password Strength: STRONG

31°C Partly sunny   ENG   04:44 PM 03-10-2023

File Edit Source Refactor Navigate Search Project Run Window Help

PasswordStrengthChecker.java ×

```java
1  import java.util.regex.*;
2
3  public class PasswordStrengthChecker {
4
5      public static void main(String[] args) {
6          String password = "MyStrongP@ssw0rd";
7          PasswordStrength strength = checkPasswordStrength(password);
8          System.out.println("Password Strength: " + strength);
9      }
10
11     public static PasswordStrength checkPasswordStrength(String password) {
12         if (password == null || password.isEmpty()) {
13             return PasswordStrength.WEAK;
14         }
15
16         int length = password.length();
17         if (length < 8) {
18             return PasswordStrength.WEAK;
19         }
20
21         if (!containsUppercase(password) || !containsLowercase(password) || !containsDigit(password) || !containsSpecialCharacter(pass
22             return PasswordStrength.MEDIUM;
23         }
24
25         return PasswordStrength.STRONG;
26     }
27
28     private static boolean containsUppercase(String password) {
29         return !password.equals(password.toLowerCase());
30     }
31
32     private static boolean containsLowercase(String password) {
33         return !password.equals(password.toUpperCase());
34     }
35
36     private static boolean containsDigit(String password) {
37         return password.matches(".*\\d.*");
38     }
39
```

Outline ×

- PasswordStrengthChecker
  - main(String[]) : void
  - checkPasswordStrength(String) : PasswordStrength
  - containsUppercase(String) : boolean
  - containsLowercase(String) : boolean
  - containsDigit(String) : boolean
  - containsSpecialCharacter(String) : boolean
  - PasswordStrength

Writable   Smart Insert   48 : 6 : 1572

31°C Partly sunny   ENG   04:45 PM 03-10-2023

File   Edit   Source   Refactor   Navigate   Search   Project   Run   Window   Help

PasswordStrengthChecker.java ×

```java
 12          if (password == null || password.isEmpty()) {
 13              return PasswordStrength.WEAK;
 14          }
 15
 16          int length = password.length();
 17          if (length < 8) {
 18              return PasswordStrength.WEAK;
 19          }
 20
 21          if (!containsUppercase(password) || !containsLowercase(password) || !containsDigit(password) || !containsSpecialCharacter(pass
 22              return PasswordStrength.MEDIUM;
 23          }
 24
 25          return PasswordStrength.STRONG;
 26      }
 27
 28      private static boolean containsUppercase(String password) {
 29          return !password.equals(password.toLowerCase());
 30      }
 31
 32      private static boolean containsLowercase(String password) {
 33          return !password.equals(password.toUpperCase());
 34      }
 35
 36      private static boolean containsDigit(String password) {
 37          return password.matches(".*\\d.*");
 38      }
 39
 40      private static boolean containsSpecialCharacter(String password) {
 41          Pattern pattern = Pattern.compile("[!@#$%^&*()_+\\-=\\[\\]{};':\",.<>?]+");
 42          Matcher matcher = pattern.matcher(password);
 43          return matcher.find();
 44      }
 45
 46      public enum PasswordStrength {
 47          WEAK, MEDIUM, STRONG
 48      }
 49  }
 50
```

Outline ×

- PasswordStrengthChecker
  - main(String[]) : void
  - checkPasswordStrength(String) : PasswordStrength
  - containsUppercase(String) : boolean
  - containsLowercase(String) : boolean
  - containsDigit(String) : boolean
  - containsSpecialCharacter(String) : boolean
  - PasswordStrength

Writable   Smart Insert   48 : 6 : 1572

31°C Partly sunny   ENG   04:45 PM   03-10-2023

Level 2

Task 3

Task: File Encryption/Decryption

Description: Create a program that encrypts or decrypts the contents of a text file using a simple encryption algorithm. Prompt the user to choose between encryption or decryption, and input the file name or path. Encrypt or decrypt the file accordingly and save the result to a new file.

Skills: File handling, string manipulation, basic input/output operations.

TicTacToe.java   FileEncryptionDecryption.java ×

```java
1  import java.io.*;
3
4  public class FileEncryptionDecryption {
5      public static void main(String[] args) {
6          Scanner scanner = new Scanner(System.in);
7
8          System.out.print("Enter 'E' for encryption or 'D' for decryption: ");
9          char choice = scanner.next().charAt(0);
10
11         if (choice == 'E' || choice == 'e') {
12             System.out.print("Enter the input file name: ");
```

Outline ×

- FileEncryptionDecryption
  - main(String[]) : void
  - encryptFile(String, String) : void
  - decryptFile(String, String) : void

Problems   @ Javadoc   Declaration   Console ×

```
<terminated> FileEncryptionDecryption [Java Application] C:\Users\as\AppData\Local\Programs\Eclipse Adoptium\jdk-17.0.3.7-hotspot\bin\javaw.exe (03-Oct-2023, 4:38:23 pm – 4:38:52 pm) [pid: 6944]
Enter 'E' for encryption or 'D' for decryption: E
Enter the input file name: task
Enter the output file name: AGRAWALRITESH
java.io.FileNotFoundException: task (The system cannot find the file specified)
        at java.base/java.io.FileInputStream.open0(Native Method)
        at java.base/java.io.FileInputStream.open(FileInputStream.java:216)
        at java.base/java.io.FileInputStream.<init>(FileInputStream.java:157)
        at java.base/java.io.FileInputStream.<init>(FileInputStream.java:111)
        at java.base/java.io.FileReader.<init>(FileReader.java:60)
        at FileEncryptionDecryption.encryptFile(FileEncryptionDecryption.java:33)
        at FileEncryptionDecryption.main(FileEncryptionDecryption.java:16)
File encrypted successfully!
```

31°C   Partly sunny   ENG   04:38 PM   03-10-2023

File   Edit   Source   Refactor   Navigate   Search   Project   Run   Window   Help

TicTacToe.java   │   FileEncryptionDecryption.java ×

```java
1  import java.io.*;
2
3
4  public class FileEncryptionDecryption {
5      public static void main(String[] args) {
6          Scanner scanner = new Scanner(System.in);
7
8          System.out.print("Enter 'E' for encryption or 'D' for decryption: ");
9          char choice = scanner.next().charAt(0);
10
11         if (choice == 'E' || choice == 'e') {
12             System.out.print("Enter the input file name: ");
```

Outline

- FileEncryptionDecryption
  - main(String[]) : void
  - encryptFile(String, String) : void
  - decryptFile(String, String) : void

Problems   @ Javadoc   Declaration   Console ×

FileEncryptionDecryption [Java Application] C:\Users\as\AppData\Local\Programs\Eclipse Adoptium\jdk-17.0.3.7-hotspot\bin\javaw.exe (03-Oct-2023, 4:38:23 pm) [pid: 6944]

```
Enter 'E' for encryption or 'D' for decryption:
```

31°C Partly sunny   ENG   04:38 PM   03-10-2023