Assignment-3 for Error Handling

Subject: CSW2 (CSE 3141)
Session: Jan to May 2025
Branch: CSE
Section: All
Course Outcomes: CO2

Learning Levels: Remembering (L1), Understanding (L2), Application (L3), and Analysis (L4).

Q no.	Questions	Learning Levels
Q1.	You are given a string containing alphanumeric characters, and your task is to design a Java program that extracts and displays the numeric characters from the given string. If no numeric characters are present, the program should display an appropriate message indicating their absence. Additionally, if the input string is **null** or empty, the program must throw a **NullPointerException** with a meaningful error message.	L1, L2
Q2.	Implement a custom exception class named **CustomNullPointerException** that replicates the behavior of the standard **NullPointerException**. However, instead of relying on default error messages or null references, this custom exception should accept a **String** message as a constructor argument. Your task is to create this custom exception class and demonstrate its usage in a Java program.	L3, L4
Q3.	Create a method that accepts a string input and converts it into an integer. Use a **try-catch** block to handle **NumberFormatException**, and if an exception occurs, prompt the user to enter a valid numeric value.	L2, L3
Q4.	Write a Java program to find the square root of an integer number. Demonstrate the use of a **try-catch** block to handle **ArithmeticException**.	L1, L2
Q5.	Demonstrate the use of a nested **try-catch** block. Write a Java program where the outer **try-catch** block handles a **NumberFormatException**, while the inner **try-catch** block handles an **ArithmeticException**.	L2, L3
Q6.	Implement a Java program that performs complex manipulations on an array of integers, including operations such as sorting, searching, and accessing elements at various indices. Introduce scenarios where accessing elements beyond the array bounds leads to an **ArrayIndexOutOfBoundsException**. Handle these exceptions gracefully to ensure the program continues execution without crashing.	L2, L3
Q7.	Design a Java program to perform matrix operations such as addition, multiplication, and transpose. Introduce scenarios where accessing elements beyond the matrix bounds results in an **ArrayIndexOutOfBoundsException**. Handle these exceptions	L3, L4

	effectively and provide meaningful error messages that clearly indicate the nature of the exception.	
Q8.	Create a custom-checked exception class named **CustomCheckedException**. Use this exception in your program to handle a specific error condition and demonstrate its usage with a **try-catch** block.	L3, L4
Q9.	Implement a method that reads an integer from the user and handles **InputMismatchException** using a **try-catch** block.	L2, L3
Q10.	Implement a Java program that reads a file path from the command-line argument and attempts to read its contents. If the file path is **null** or points to a non-existent file, throw a custom **FileNotFoundException**. If the file exists but cannot be read due to permission issues, throw a custom **FileReadPermissionException**. Your task is to create these custom exception classes and handle them appropriately in your program.	L3, L4
Q11.	Write a program that reads data from a file and performs some processing. Handle checked **IOException** by using **try-catch** block to catch and handle the exception.	L3, L4
	-END-	