

Hands-on No. : 4

Topic : Exception Handling and Collections

Date : 05.08.2025

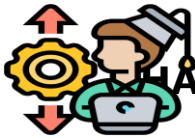
Solve the following problems

Question No.	Question Detail
1	<p>Develop a Calculator application that performs the following basic arithmetic operations:</p> <p>Addition</p> <p>Subtraction</p> <p>Multiplication</p> <p>Division</p> <p>Requirements:</p> <p>The application should accept only numeric inputs.</p> <p>Inputs containing characters or strings should not be accepted.</p> <p>For division, the denominator must not be zero.</p> <p>For multiplication, neither the multiplier nor the multiplicand should be 0 or 1.</p> <p>Exception Handling:</p> <p>Create and use user-defined exceptions to handle invalid inputs and violations of the above conditions.</p> <p>Display appropriate error messages when an exception occurs and ensure the program continues or exits gracefully.</p> <p>Sample Input/Output</p> <p>Enter first number: 10</p> <p>Enter second number: 20</p> <p>Enter operation (+, -, *, /): +</p> <p>Result: 30.0</p> <p>Enter first number: ten</p> <p>Enter second number: 20</p>

It is going to be hard but, hard does not mean impossible.

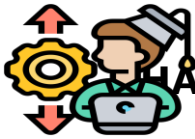
	<p>Enter operation (+, -, *, /): +</p> <p>Error: Invalid input. Only numeric values are allowed.</p> <p>Enter first number: 50</p> <p>Enter second number: 0</p> <p>Enter operation (+, -, *, /): /</p> <p>Error: Division by zero is not allowed.</p> <p>Enter first number: 0</p> <p>Enter second number: 5</p> <p>Enter operation (+, -, *, /): *</p> <p>Error: Multiplication with 0 or 1 is not allowed.</p>
2	<p>Create a simple Bank Transaction Application that allows a student to perform either a deposit or a withdrawal, based on their choice. The application must handle invalid input scenarios using a user-defined exception.</p> <p>Requirements:</p> <ul style="list-style-type: none"> The user should input the type of transaction: "deposit" or "withdraw". The system should ask for the transaction amount. If the transaction is "withdraw" and the amount exceeds the current balance, a custom exception InvalidTransactionException should be thrown. If the transaction amount entered is zero or negative, the same custom exception should be thrown. Ensure that the exception is caught and an appropriate message is displayed. <p>Note:</p> <ul style="list-style-type: none"> Initialize the account with a fixed balance (e.g., ₹5000). You may use a single custom exception InvalidTransactionException for all invalid conditions. <p>Sample Input and Output</p>

It is going to be hard but, hard does not mean impossible.



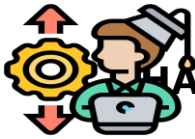
	<p>Enter transaction type (deposit/withdraw): withdraw</p> <p>Enter amount: 6000</p> <p>Transaction Failed: Withdrawal amount exceeds current balance.</p> <p>Enter transaction type (deposit/withdraw): deposit</p> <p>Enter amount: -100</p> <p>Transaction Failed: Amount must be greater than zero.</p>
3	<p>Write a program that accepts a string input from the user and counts how many times each character appears in the string.</p> <p>Use a HashMap to store:</p> <p>Key: character</p> <p>Value: number of times the character appears</p> <p>Ignore spaces and consider both uppercase and lowercase letters as the same (optional based on requirement).</p> <p>Finally, display each character along with its count.</p> <p>Sample Input and Output</p> <p>Enter a string: classroom chaos</p> <p>c: 2</p> <p>l: 1</p> <p>a: 2</p> <p>s: 3</p> <p>r: 1</p> <p>o: 2</p> <p>m: 1</p> <p>h: 1</p>
4	<p>Write a Java program that allows the user to enter a list of names. First, ask the user how many names they would like to input. Then, read each name and store them in an ArrayList.</p> <p>After all the names have been entered, display only those names whose length is greater than 5 characters, effectively removing names that are 5 characters or fewer.</p>

It is going to be hard but, hard does not mean impossible.



	<p>SampleInput</p> <p>How many names: 4</p> <p>Enter name 1: Elvis</p> <p>Enter name 2: Priscilla</p> <p>Enter name 3: Vernon</p> <p>Enter name 4: LisaMarie</p> <p>Sample Output:</p> <p>Priscilla</p> <p>Vernon</p> <p>LisaMarie</p>
5	<p>Write a Java program that reads a list of words from the user and stores them in a TreeSet (to keep them sorted and unique). The program should offer a menu-driven interface that lets the user choose from several operations to perform on the list of words.</p> <p>Program Requirements:</p> <ul style="list-style-type: none">• Ask the user how many words they want to enter.• Read each word and store it in a <code>TreeSet<String></code>.• Display a menu with the following operations. Each option must be handled using a separate function: <p>Menu Options:</p> <ol style="list-style-type: none">1. Display All Words Show all the words in the order maintained by <code>TreeSet</code> (i.e., sorted alphabetically).2. Display Words with Capitalized Plurals Display all words. If a word ends with the letter 's' (indicating a possible plural), show it in uppercase. All other words remain unchanged.3. Display list of Words in Reverse Order Show all the stored list of words in reverse order4. Display Words Excluding Plurals Display only the words that do not end with the letter 's' (i.e., exclude all words that end with 's').

It is going to be hard but, hard does not mean impossible.



5. Exit

End the program.

Sample Input and Output

Enter number of words: 6

Enter word 1: cats

Enter word 2: apple

Enter word 3: bananas

Enter word 4: desk

Enter word 5: apples

Enter word 6: chair

Menu:

1. Display All Words
2. Display Words with Capitalized Plurals
3. Display Words in Reverse Order
4. Display Words Excluding Plurals
5. Exit

Enter your choice: 1

apple

apples

bananas

cats

chair

desk

Enter your choice: 2

apple

APPLES

BANANAS

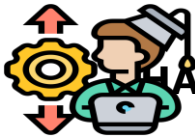
CATS

chair

desk

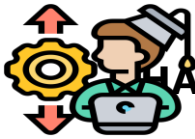
Enter your choice: 3

It is going to be hard but, hard does not mean impossible.



	<p>desk</p> <p>chair</p> <p>cats</p> <p>bananas</p> <p>apples</p> <p>apple</p> <p>Enter your choice: 4</p> <p>Output:</p> <p>apple</p> <p>chair</p> <p>desk</p>
6	<p>Julie has created a program that uses a HashMap to store the names of students as keys and their marks as values.</p> <p>Your task is to write a program using user-defined functions to perform the following operations:</p> <p>Requirements:</p> <ol style="list-style-type: none">1. Filter and Push to Stack Create a function that goes through the HashMap and pushes the names of students into a stack only if their marks are greater than 75.2. Pop and Display from the filtered Stack Create another function that pops each name from the stack and displays it on the screen. <p>Sample Input and Output:</p> <p>HashMap:</p> <pre>{ "Alice": 82, "Bob": 67, "Charlie": 91, "Diana": 74, "Ethan": 88 }</pre>

It is going to be hard but, hard does not mean impossible.



	After filtering the output from the stack, Ethan,Charlie, Alice
7	<p>Create a class named Person to represent a person with the following details: firstName (String) lastName (String)</p> <p>Now write a driver program to:</p> <ul style="list-style-type: none">• Ask the user how many people they want to enter.• Accept the first and last names of n people.• Store all the Person objects in a suitable collection (e.g., ArrayList).• Sort the list of persons lexicographically by their first name.• Display the sorted list in the format: FirstName LastName

It is going to be hard but, hard does not mean impossible.