

Java Exception Handling Lab Exercises

1. Write a program that asks the user to input two integers and divides the first by the second. Catches any division by zero and prints an error message.
2. Write a program that ask for two integers from the user and catch both `ArithmeticException` and `InputMismatchException`.
3. Modify question 1 to include a finally block that prints "Operation completed." regardless of the outcome.
4. Create a custom exception `InvalidAgeException`. In a method `validateAge(int age)`, throw `InvalidAgeException` if `age < 18` and In `main()`, get age from user and validate.
5. Write a Java program to create an array of size 5 and try accessing index 10. Catch and handle the exception.
6. Write a Java program to create a method that takes an integer as a parameter and throws an exception if the number is odd.
7. Write a Java program that reads a file and throws an exception if the file is empty.
8. Write a Java program to create a method that takes a string as input and throws an exception if the string does not contain vowels.
9. Write a Java program that reads a list of numbers from a file and throws an exception if any of the numbers are positive.
10. Write a Java program that reads an integer input. Handle the case when the user enters invalid input.
11. Create three methods: `methodA()`, `methodB()`, and `methodC()`.
 - `methodC()` throws an `ArithmeticException`.
 - `methodB()` calls `methodC()` without handling it.
 - `methodA()` calls `methodB()` and handles the exception
12. Ask the user to enter a number. Convert it to int using `Integer.parseInt()`. Catch and handle `NumberFormatException`.
13. Create a `BankAccount` class with:
 - balance
 - `withdraw(double amount)` method

if the user tries to withdraw more than the balance, throw a custom exception `InsufficientFundsException`.

14. A university system processes student marks to validate student marks and handle exceptions.

Requirements:

- Accept marks (0–100) from the user.
- Throw `InvalidMarksException` if marks are out of range.
- Compute grade if input is valid.

15. An e-commerce app lets users place orders to Handle input and business logic exceptions.

Requirements:

- Ask for product quantity.
- If quantity is negative or zero, throw `InvalidQuantityException`.
- If quantity exceeds available stock (e.g., 10 units), throw `OutOfStockException`.

16. A login system validates username and password to securely handle invalid login attempts.

Requirements:

- Accept username and password.
- If credentials are incorrect, throw `AuthenticationFailedException`.
- Use exception messages to guide the user.

Java File Handling Lab Exercises

1. Write a Java program that creates a file named output.txt and writes the text "Hello, File Handling in Java!" into it. Handle any exceptions that may occur during file operations.
2. Write a Java program that reads the contents of the file output.txt and displays the text on the console. Handle exceptions related to file reading.
3. Write a Java program that appends the line "This is an appended line." to the file output.txt without overwriting its existing content.
4. Write a Java program that checks the properties of the file output.txt. Display:
 - File name
 - Absolute path
 - Whether the file is readable
 - Whether the file is writable
 - File size in bytes
5. Write a Java program to delete the file output.txt if it exists. Display a message indicating whether the deletion was successful or if the file was not found.
6. Write a Java program to create a new directory named myFiles in the current working directory. Display whether the directory creation was successful.