Assignment 1

Title: Handle Arithmetic Exception

Problem Statement:
Write a Java program that accepts two integers and performs division. Handle the scenario where the denominator is zero.

Requirements / Constraints:

- Use try-catch
- Catch ArithmeticException
- Display a user-friendly message

Expected Outcome:
Program should not crash and should display "Division by zero is not allowed".

Assignment 2

Title: Handle Array Index Exception

Problem Statement:
Create an array of size 5 and try to access an invalid index.

Requirements / Constraints:

- Use try-catch
- Catch ArrayIndexOutOfBoundsException

Expected Outcome:
Program should print "Invalid array index accessed".

Assignment 3

Title: Handle Null Pointer Exception

Problem Statement:
Create a string variable with null value and attempt to call a method on it.

Requirements / Constraints:

- Use try-catch
- Catch NullPointerException

Expected Outcome:
Program should handle the exception gracefully.

Assignment 4

Title: Multiple Catch Blocks

Problem Statement:
Write a program that can throw both ArithmeticException and ArrayIndexOutOfBoundsException.

Requirements / Constraints:

- Use multiple catch blocks
- Each exception should have a specific message

Expected Outcome:
Correct exception should be caught and message displayed.

Assignment 5

Title: Exception Handling with finally Block

Problem Statement:
Write a program that opens a file and ensures the file resource is closed using finally.

Requirements / Constraints:

- Use try-catch-finally
- Simulate file handling logic

Expected Outcome:
finally block should always execute.

Assignment 6

Title: Nested Try-Catch

Problem Statement:
Demonstrate nested try-catch blocks for handling different exceptions at different levels.

Requirements / Constraints:

- Inner try-catch for ArithmeticException
- Outer try-catch for generic exception

Expected Outcome:
Proper exception handling at each level.

Assignment 7

Title: User Input Validation Using Exception

Problem Statement:
Accept user input for age and validate that age must be greater than 18.

Requirements / Constraints:

- Throw an exception if age < 18
- Handle the exception properly

Expected Outcome:
Program should display "Age not eligible".

Assignment 8

Title: Custom Exception Creation

Problem Statement:
Create a custom exception called InvalidBalanceException for a bank application.

Requirements / Constraints:

- Extend Exception
- Throw exception when balance < minimum balance

Expected Outcome:
Custom exception message should be displayed.

Assignment 9

Title: Exception Propagation

Problem Statement:
Write a program where an exception occurs in one method and is propagated to the calling method.

Requirements / Constraints:

- Use throws keyword
- Handle exception in main method

Expected Outcome:
Exception should propagate correctly and be handled.

Assignment 10

Title: Re-Throwing an Exception

Problem Statement:
Catch an exception, log a message, and rethrow it to the calling method.

Requirements / Constraints:

- Use throw
- Preserve original exception details

Expected Outcome:
Both log and exception handling should occur.

Assignment 11

Title: Exception Handling in Method Overriding

Problem Statement:
Demonstrate rules of exception handling in method overriding using parent and child classes.

Requirements / Constraints:

- Child class should not throw broader checked exceptions

Expected Outcome:
Program should compile and run successfully.

Assignment 12

Title: Exception Handling with Custom Validation Layer

Problem Statement:
Design a service method that validates input and throws different custom exceptions based on validation failure.

Requirements / Constraints:

- Use at least two custom exceptions
- Handle them in a controller/main class

Expected Outcome:
Different error messages based on failure type.