Web Technologies Lab

Lab 04 **Marks 100**

Instructions

Work on this lab individually.

You are **NOT** allowed to use the internet, or mobile phone.

You are **NOT** allowed to borrow anything from your peer student.

What you have to do

Program the following tasks. The name of your files will be according to the task given in this lab.

Task 1 [50]

A bank wants a Java program to manage bank accounts where users can deposit and withdraw money. However, certain conditions must be handled using exceptions.

Instruction:

1. Create a custom exception **`InsufficientFundsException`**, which should be thrown when a withdrawal amount exceeds the available balance.
2. Implement a class **`BankAccount`**
   - with attributes:
     i. accountNumber
     ii. balance
   - Also provide methods:
     i. **`deposit(double amount):`** Adds money to the balance.
     ii. **`withdraw(double amount):`** Deducts money but throws **`InsufficientFundsException`** if the balance is insufficient.
3. In the **main method**, do the following:
   - Ask the user to enter an amount to withdraw.
   - Use try-catch-finally to handle the custom exception and display an appropriate message.
   - Ensure the **`finally`** block prints `"Transaction complete."`, whether an exception occurs or not.
4. Use throws for **`withdraw(double amount)`** to indicate it may throw an exception.

Task 2 [50]

A university needs a Java program that takes a student's marks as input and assigns grades based on the score. However, invalid marks (negative or above 100) should trigger a **custom exception**.

Instruction:

1. Create a custom exception **`InvalidMarksException`** that is thrown when marks are not in the range 0-100.
2. Implement a class **`Student`**
   - with attributes:
     i. name
     ii. marks
   - Also provide method **`assignGrade():`**
     i. Throws **`InvalidMarksException`** for invalid marks.
     ii. Returns **`"A"`** for marks ≥ 85, **`"B"`** for 70-84, **`"C"`** for 50-69, **`"F"`** otherwise.

3. In the **main method**, do the following:
    - Take user input for a student's marks.
    - Use try-catch-finally to handle the exceptions.
    - Print **`"Grade assigned successfully."`** in the finally block, regardless of success or failure.

4. Use throws for **`assignGrade()`** to indicate it may throw an exception.

☺ ☺ ☺ **BEST OF LUCK** ☺ ☺ ☺
