Q.You are building a banking system where users can withdraw money. If a user tries to withdraw more money than they have in their balance, a custom exception named InsufficientBalanceException should be thrown.

Task:

1. Create a custom exception class named InsufficientBalanceException.

2. Create a class BankAccount that has a private balance (initialized to 1000).

3. Implement a method withdraw(int amount) in the BankAccount class that checks the balance and throws the InsufficientBalanceException if the withdrawal amount exceeds the balance.

4. Write a main method to simulate a withdrawal scenario, where the user attempts to withdraw more money than their balance and catches the exception.

Q. Write a simple Java program that creates two threads. Each thread should print numbers from 1 to 100 along with the thread name.

- Implement by extending Thread.
- Implement by implementing Runnable interface.

Q. Synchronization with Threads

You are simulating a shared printer in a computer lab. Multiple users (represented by threads) are trying to print documents using the same printer. However, the printer can only print one document at a time. If multiple users try to print simultaneously, their outputs may get mixed. To avoid this, we must use synchronization.

Your Tasks:

1. Create a Printer class with a method printDocument(String userName) that prints 5 lines simulating a document being printed.

    - Use a for loop and Thread.sleep(500) to simulate delay between lines.

    - Ensure the method is synchronized so only one thread can use it at a time.

2. Create a User class that extends Thread:

    - It should take a Printer object and a user name.

    - In the run() method, call printer.printDocument(userName);

Q.Write a Java program to copy the contents of one text file to another using either byte streams and character streams

Q. Write a Java program using RandomAccessFile to do the following:

- Write the string "Hello World" to a file named data.txt.

- Move the file pointer to the beginning.

- Read and display the content from the file.

Q.Serialization and Deserialization using ObjectInputStream and ObjectOutputStream

- Create a class named Student with id and name fields.
- Serialize a Student object and save it to a file named student.ser.
- Then read the object back from the student.ser file and display the values.