**IMPLEMENTATION OF EXCEPTIONS**

**AIM:**

Write a java program that uses the try-catch-finally block to handle exceptions during the bank transfer. An exception should be thrown when the transfer amount exceeds the available balance in the customer's account.

**PROGRAM**:

/\*\*

\*

\* @author 2162014

\*/

public class excep {

private double balance;

public excep(double initialBalance) {

balance = initialBalance;

}

public void transfer(double amount, excep recipient) throws

InsufficientFundsException {

try {

if (amount > balance) {

throw new InsufficientFundsException("Transfer amount exceeds available balance");

} else {

balance -= amount;

recipient.balance += amount;

System.out.println("Transfer successful!");

}

} catch (InsufficientFundsException e) {

System.out.println("Transfer failed: " + e.getMessage());

throw e;

}

}

public static void main(String[] args) {

excep account1 = new excep(1000.0);

excep account2 = new excep(500.0);

try {

System.out.println("Acc1 balance before transfer: " + account1.balance);

System.out.println("Acc2 balance before transfer: " + account2.balance);

account1.transfer(600.0, account2);

// account2.transfer(400.0, account1);

} catch (InsufficientFundsException e) {

// Handle the exception here

} finally {

System.out.println("Acc1 balance after transfer: " + account1.balance);

System.out.println("Acc2 balance after transfer: " + account2.balance);

}

}

}

class InsufficientFundsException extends Exception {

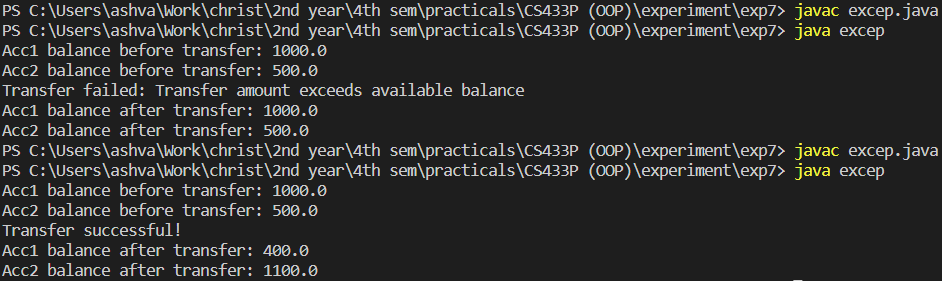
public InsufficientFundsException(String message) {

super(message);

}

}

**OUTPUTS**:

****

**RESULTS:**

The java program was created successfully to demonstrate the use of the try-catch-finally block to handle exceptions during the bank transfer