Lab 11 : Build a **Customized exception** in the name of **DivisionByZeroException** which will be invoked when you try to divide a number by zero

```
// Custom exception class
class DivisionByZeroException extends Exception {
    public DivisionByZeroException(String message) {
        super(message);
    }
}

public class CustomExceptionDemo {
    // Method to perform division and throw custom exception if denominator is zero
    static double divide(int numerator, int denominator) throws DivisionByZeroException {
        if (denominator == 0) {
            throw new DivisionByZeroException("Cannot divide by zero!");
        }
        return (double) numerator / denominator;
    }

    public static void main(String[] args) {
        int numerator = 10;
        int denominator = 0;

        try {
            double result = divide(numerator, denominator);
            System.out.println("Result of division: " + result);
        } catch (DivisionByZeroException e) {
            System.out.println("Exception caught: " + e.getMessage());
        } finally {
            System.out.println("Finally block executed");
        }
    }
}
```

o/p

```
Exception caught: Cannot divide by zero!
Finally block executed
```

Exercise : Create a custom exception called **InsufficientBalanceException** to represent a situation where a user tries to withdraw more money than is available in their bank account. (**refer to the Class notes Module 4**)