Lesson –12 - Exception Handling

1. Write your program to get the input of score in the range of 0 to 100 from the console. Apply the necessary Exception handling to avoid negative, non-numbers. [Use API Exception only].

   Handle the following,

   - InputMismatchException for the wrong entry of data instead of int value.
   - UnsupportedOperationException for the inputs not in the range of 0-100.

2. Create a class called CustomerAccount with the following attributes and methods.

Attributes: Cus_name, Acc_No, Balance.

Implement the below methods

public boolean deposit (double amount)

Used to increase the balance. If successfully deposited return true, incase of negative inputs return false.

public boolean withdraw(double amount)

Used to reduce the balance and return true for successful withdraw or else return false.

public double getBalance();

Used to return the current balance.

Create a user defined/custom exception to handle the following situation for the above problem.

1. Withdraw amount exceed the balance
2. Balance reach below 100$.

3. Take the data structures problem you worked and integrate the exception handling mechanism you learned. It will be a best practice.

Example

- If the Queue is empty can throw the Custom exception.
- In the MyStringList.java file, if the index is not in the valid range throw IndexOutOfBoundsException

Likewise apply the suitable exceptions and practice with try catch, throw, throws, and Custom Exceptions.