Development Task

Prior to your technical interview at XXX, we would like you to implement a small test application. Reviewing your implementation and discussing how it can be enhanced will form part of the interview.

The task is to implement a class that builds an order book from a stream of orders and prints out the 'by level' order book at the end of a run. A detailed specification of a 'by level' order book is provided on page 2 of this document.

A simple task framework and skeleton solution have been produced, and you are required to focus on implementing the OrderConsumer class.

You should:

1. Initialise your structures in StartProcessing()
2. Build the 'by level' order books by processing the order events in HandleOrderAction().
3. Write the 'by level' order books to the ILog interface in FinishProcessing().
4. Implement a basic logger that writes your output to the console.

A few important notes:

1. Order books are built for each individual symbol. It is a mistake to aggregate orders for different symbols in the same order book.
2. You must maintain the 'by level' order books in real time and not wait until the FinishProcessing() event to build them.
3. A typical order book might have anywhere from 5 to 500 price levels at any time, and a typical order might have 4 or more modifications applied to it before being removed.
4. A sample stream of orders is provided in the AppEnvironment class.

We would expect candidates to spend no more than 4 hours on their submission.

Please send the completed solutions in an e-mail with attached zip file to your agency.

Order Book Definition

Many markets are order driven, which is to say that traders submit orders to buy and sell instruments for a given price and volume.

This exercise assumes a market in which you are only notified of market orders. These are a firm request to buy or sell a certain quantity of the instrument (stock or option or future) for no more than the given 'limit price'.

For example, an order to buy 100 Microsoft shares for 28.70 means that the trader wants to buy 100 MSFT shares and will not pay more than $28.70. Multiple traders place individual orders at different prices and sizes, leading to the build up of an order book. The 'by level' order book shows the number of order and their total volume (i.e. the cumulative quantity) at a given price level.

Order may be removed (either because a trader has cancelled an order or because it has traded) and the price and/or volume may be edited.

Definitions:

- Bid Price - The highest price that a trader is willing to pay to buy an instrument
- Bid Size - The number of contracts (or shares) that are available at the bid price.
- Ask Price - The lowest price that a trader is willing to accept to sell an instrument
- Ask Size - The number of contracts (or shares) that are available at the ask price.
- Order book level - a tuple containing a price, size and order count.

The order book includes all buy and sell orders that are currently pending for an instrument. It contains two 'sides' - bid and ask. Each side consists of multiple levels that are aggregated based on the price, and is sorted such that the level that has highest bid is on the top of the bid side, and the lowest ask is on the top of the ask side.

Here is a sample order book for a specific instrument:

Bid Price	Bid Size	Order Count
22.00	7	1
21.00	17	2
19.00	12	2

Ask Price	Ask Size	Order Count
10.40	26	2
15.14	10	1